HP64000 Logic Development System

Model 64601A Logic Timing Analyzer Reference Manual

HEWLETT
PACKARD

CERTIFICATION

Hewlett-Packard Company certifies that this product met its published specifications at the time of shipment from the factory. Hewlett-Packard further certifies that its calibration measurements are traceable to the United States National Bureau of Standards, to the extent allowed by the Bureau's calibration facility, and to the calibration facilities of other International Standards Organization members.

WARRANTY

This Hewlett-Packard system product is warranted against defects in materials and workmanship for a period of 90 days from date of installation. During the warranty period, HP will, at its options, either repair or replace products which prove to be defective.

Warranty service of this product will be performed at Buyer's facility at no charge within HP service travel areas. Outside HP service travel areas, warranty service will be performed at Buyer's facility only upon HP's prior agreement and Buyer shall pay HP's round trip travel expenses. In all other cases, products must be returned to a service facility designated by HP.

For products returned to HP for warranty service. Buyer shall prepay shipping charges to HP and HP shall pay shipping charges to return the product to Buyer. However, Buyer shall pay all shipping charges, duties, and taxes for products returned to HP from another country.

HP warrants that its software and firmware designated by HP for use with an instrument will execute its programming instructions when properly installed on that instrument. HP does not warrant that the operation of the instrument, or software, or firmware will be uninterrupted or error free.

LIMITATION OF WARRANTY

The foregoing warranty shall not apply to defects resulting from improper or inadequate maintenance by Buyer, Buyer-supplied software or interfacing, unauthorized modification or misuse, operation outside of the environmental specifications for the product, or improper site preparation or maintenance.

NO OTHER WARRANTY IS EXPRESSED OR IMPLIED. HP SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

EXCLUSIVE REMEDIES

THE REMEDIES PROVIDED HEREIN ARE BUYER'S SOLE AND EXCLUSIVE REMEDIES. HP SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER BASED ON CONTRACT, TORT, OR ANY OTHER LEGAL THEORY.

ASSISTANCE

Product maintenance agreements and other customer assistance agreements are available for Hewlett-Packard products.

For any assistance, contact your nearest Hewlett-Packard Sales and Service Office.

Logic Timing Analyzer
Reference Manual

© COPYRIGHT HEWLETT-PACKARD COMPANY 1982, 1983

LOGIC SYSTEMS DIVISION
COLORADO SPRINGS, COLORADO, U.S.A.

ALL RIGHTS RESERVED

Printing History

Each new edition of this manual incorporates all material updated since the previous edition. Manual change sheets are issued between editions, allowing you to correct or insert information in the current edition.

The part number on the back cover changes only when each new edition is published. Minor corrections or additions may be made as the manual is reprinted between editions. Vertical bars in a page margin indicate the locations of reprint corrections.

First Printing............June 1982 (P/N 64601-90903)
Second Edition..............July 1983 (P/N 64601-90905)

Table of Contents

Chapter 1: General Information

Chapter 2: Installation

Chapter 3: Getting Started With The Timing Analyzer

Chapter 4: Utility Keys

Table of Contents (Cont'd)

Chapter 4: Utility Keys (Cont'd)

Chapter 5: Format Specification

Chapter 6: Trace Specification

Chapter 7: Timing Diagram Display Specification

Chapter 8: Trace List Display Specification

Table of Contents (Cont'd)

Chapter 9: Post_Process Specification

Chapter 10: Interactive Measurements

Chapter 11: Theory of Operation

List of Illustrations

List of Tables

Chapter 1

GENERAL INFORMATION

INTRODUCTION

This manual describes how to operate the Hewlett-Packard Model 64601A Logic Timing Analyzer. This chapter describes the physical and electrical characteristics of the analyzer. At the end of this chapter is the table of timing analyzer specifications guaranteed by HP.

Chapter 2 contains instructions and information to help you install the logic timing analyzer into a mainframe. This chapter also shows how to make user copies of software supplied on flexible discs.

Chapter 3 of this manual provides a set of procedures to help you become familiar with the logic timing analyzer. As soon as you can, you should perform the procedures in chapter 3.

Chapter 4 discusses the utility keys that are used throughout the timing analyzer, and describes their functions.

Chapters 5 through 9 discuss the specifications and displays of the analyzer and show how to use these to make the types of measurements desired.

Chapter 10 discusses how to make measurements using the interactive trigger and enable controls available in measurement systems composed of the logic timing analyzer and other Model 64000 analyzer modules.

Chapter 11 discusses technical aspects of timing analyzer measurements.

Appendices in this manual show the operating syntax of the complex softkeys, list and describe the operations performed by each of the softkeys, list all of the error and status messages displayed by the timing analyzer, and list and define each of the special symbols used on timing analyzer displays.

INSTRUMENT DESCRIPTION

The timing analyzer subsystem consists of a 200-MHz control board, and either one or two 200-MHz data acquisition boards (each with 8-channels of sophisticated active probing). These boards are combined to form two basic subsystems: an 8-channel subsystem using one data acquisition board, or a 16-channel subsystem using two data acquisition boards.

The timing analyzer board assemblies are designed to be installed in any of the 64000 system mainframes (except the 64100A mainframes with serial number prefixes below 2149A). Multiple subsystems can operate in a single mainframe, as follows:

a. 64110A mainframes can accept one 16-channel timing subsystem and one 8-channel subsystem occupying 5 card slots.

b. 64100A mainframes can accept three 16-channel timing subsystems occupying 9 card slots.

Operating software for the timing analysis subsystem can reside either on a hard disc or on local mass storage (dual flexible discs). Since software is not resident in ROM, the operating system can easily be enhanced or updated.

MODES OF OPERATION

The timing analyzer offers four modes of operation. These modes are described, briefly, below.

1. Wide Sample Mode - The most versatile and, therefore, the most used mode. Provides data acquisition on 8 or 16 channels (depending on the subsystem) from 2 Hz to 200 MHz with a memory depth of 4060 samples.

2. Fast Sample Mode - Provides added resolution with a fixed sample rate of 400 MHz and memory depth of 8140 samples using one-half the channels in the subsystem (4 or 8 channels).

3. Glitch Capture Mode - Captures glitches between samples of 3 ns, or wider, using one-half the number of channels in the subsystem (4 or 8 channels).

4. Dual Threshold Mode - Captures data relative to two thresholds instead of one. Resulting display is a three-level trace using one-half the number of channels in the subsystem (4 or 8 channels). This mode is used for investigating tristating, loading, bus contention, or rise and fall times.

FINDING EVENTS IN THE DATA MEMORY

The timing analyzer has a "find" feature that automatically searches the entire memory to find samples that meet specifications of your choice. The timing analyzer can find where a specific pattern first began or ended, or where the pattern had remained stable for a selected time duration. The time duration can be a time greater than or a time less than a preset period.

MARKING EVENTS IN THE DATA MEMORY

The timing analyzer has a "mark" feature that allows automatic measurement of an interval by simply defining the states found at the start and end the interval, and assigning mark_x and mark_o to identify the samples that meet these definitions. The timing analyzer will display the measurement of the interval between the two marked samples.

While mark_x and mark_o can only appear one time each in the data memory, four other marks mark_<abcd> are available to be defined. The mark_<abcd> marks will appear at all of the points in memory that meet their definitions. These four marks can be used to count the number of selected events occurring within any interval defined by mark_x and mark_o.

MAKING STATISTICAL MEASUREMENTS

The analyzer can make statistical measurements to characterize intervals whose boundries are mark_x and mark_o. While making a series of measurements, the timing analyzer can count the occurrences of selected events (marked by mark_<abcd>). It can compute the mean count of events and the standard deviation in the count of events for an interval. It can also record the minimum and maximum counts of events found during several measurements of the interval.

TRIGGER CAPABILITIES

The timing analyzer offers real-time trigger recognition on pattern durations, pattern transitions, and pattern sequences between pod 1 and pod 2 (in 16-channel subsystems). The post processing capability of the timing analyzer can be used to extend triggering. During data acquisition, emphasis is placed on sample resolution, not on recognizing the occurrences of complex events. The data in memory can be processed after the trace is complete to see if a complex event has been captured. If it is found in the captured data, the data can be retained for analysis. If it is not found, the data can be purged and a new trace begun, automatically. The conditions that can be identified to extend triggering are:

a. Time intervals (periods between the mark_x and mark_o samples) that lasted longer than or shorter than a selected period.

b. Too many or too few occurrences of mark_<abcd> events during a time interval between the mark_x and mark_o samples.

c. Marked events being found in a specific order or being found "not" in a specific order.

DISPLAYING STATE FLOW

The timing analyzer can be used to capture and display state flow from a very fast system. State flow is usually captured synchronously by using a clock from the system under test. The timing analyzer samples data asynchronously. By sampling data at a much faster rate than the state flow rate in the system under test, the post-process capability of the timing analyzer can be used to compose a list of state flow in the following ways:

a. The timing analyzer can list samples where states remained stable for enough time to satisfy setup and hold times in the system under test.

b. By connecting one of the sampling channels to the system clock, the timing analyzer can post-process a list of samples that were obtained just before or just after the "read-data" transition in the system under test.

c. The four mark_<abcd> specifications can be used to identify characteristics of states in memory. The timing analyzer can compose a list of marked samples. You can include names in the mark specifications and obtain a column in the list that shows descriptive mark names.

COMPARING TRACES ON A DIAGRAM

The timing analyzer can present a diagram composed of traces obtained from two sources: the present data in memory and any stored file defined as a reference source. You can arrange the traces in any order. This capability allows quick visual comparisons of traces obtained at different times from the same measured point to see if the required similarity exists between the traces. This capability should not be used for bit-to-bit comparisons because asynchronous sampling allows some mismatch in the capture of data and recognition of trigger.

EXECUTING A MEASUREMENT WITH COMPLEX SPECIFICATIONS

The total measurement, along with all specified post processing, is performed automatically as a single data acquisition cycle in response to the "execute" command.

TIMING ANALYZER SPECIFICATIONS

Timing analyzer specifications are listed in table 1-1.

Table 1-1. Timing Analyzer Specifications

RESOLUTION

Total Skew From Probe Tip
Within Pod: plus or minus 1.5 ns.
Pod to Pod: plus or minus 3.0 ns.
Conditions True for Input Signal
V_H = minus 1.0V.
V_L = minus 1.6V.
V_{TH} = minus 1.3V.
Slew Rate = less than 0.25V per ns.

Sample Rate Accuracy: approximately plus or minus 0.002%, rate adjustable from 2 Hz to 400 MHz (in fast sample mode) in 1, 2, 4 sequence.

PROBE CHARACTERISTICS

Input Z: 100K ohms plus or minus 2%, shunted by less than 6 pF.

Dynamic Range: plus or minus 10V.

Maximum Input: plus or minus 40V.

Threshold Accuracy: plus or minus 50 mV or plus or minus 2%, whichever is greater.

Hysteresis: approximately 50 mV.

Drive Requirements
Minimum Input Amplitude: 600 mV peak-to-peak.
Minimum Input Overdrive: 200 mV or 25% of input amplitude, whichever is greater.
Minimum Pulse Width: 3 ns at threshold.

GLITCH MODE

Maximum Sample Rate: 100 MHz.

Minimum Width: 3.0 ns at threshold.

Maximum Width: sample period less 4.0 ns.

Table 1-1. Timing Analyzer Specifications (Cont'd)

TRIGGERING

Time Duration Accuracy: plus or minus 20% plus 2 ns.

Minimum Width for "Narrower Than" Trigger: approximately 6 ns.

Minimum Width for Transition Trigger: approximately 6 ns.

Displayed Position Accuracy
Wide, Glitch, and Dual Threshold Modes: plus or minus 4 samples.
Fast Sample Mode: plus or minus 8 samples.

Delay from Input to External BNC Drive: approximately 60 ns.

Delay from Input to Internal IMB Drive: approximately 55 ns.

Dead Time for Restart Measurement Reset: approximately 50 ns plus the time required to fill the memory with the selected amount of pretrigger information.

Time Duration Restart: for accurate restarts, pattern must go false for at least 1.5 times the selected time duration.

BNC DRIVE (4) DRIVE SPECIFICATIONS

Output Signal Swing in Transition Trigger Mode
Amplitude: approximately 2.0V with 50-ohm load.
Width at 50%: approximately 10 ns.

Output Signal Swing in "Width Greater Than" Trigger Mode
Amplitude: approximately 2.5V.
Width: input trigger width less selected duration.

Output Signal Swing in "Width Less Than" Trigger Mode
Amplitude: same as in transition trigger mode.
Width: same as in transition trigger mode.
Position: occurs when trigger pattern disappears, before selected duration times out.

Chapter 2

INSTALLATION

INTRODUCTION

The purpose of this section is to provide the information needed to install and interconnect the timing analysis boards in a mainframe and to connect the probe cables to make measurements.

Along with the electronic hardware, you receive a set of flexible discs or a tape. These contain the software that you need to operate the timing analyzer. If you receive flexible discs, you should make user copies of these discs to operate the analyzer. Save the original set of flexible discs as a master set. The procedures for making these copies are also described in this section.

INSTALLATION

The timing analyzer is made up of a timing control board and either one or two timing acquisition boards. There are certain requirements of board installation which must be followed no matter which configuration (one or two acquisition boards) you are installing. One of these requirements is the order in which the boards are installed. Always observe the following order of installation:

CAUTION

Make sure power is off prior to installing boards.

1. Install one acquisition board in the slot with the lowest available number in the mainframe.

2. Install the control board next to the acquisition board, in the next higher numbered slot.

NOTE

Observe the number of the slot where you install the control board. This number will appear on the softkey label line in the monitor or measurement systems display beside the word timing (i.e.: timing_4). The number identifies the timing analyzer with a particular slot so that two timing analyzers can be installed in the same mainframe and the unique identities of each can be maintained.

3. If another acquisition board is to be used, install it next to the control board, in the next higher numbered slot. The slot/board configuration at this point should be: lowest numbered slot contains one acquisition board, next higher numbered slot contains the control board, and the next higher numbered slot contains the second acquisition board.

4. Connect the ribbon cable between the connectors on the left side (when viewed from the rear of the mainframe) of the control and acquisition board(s). The control board connector is keyed to ensure proper connection of the cable.

CAUTION

> Coaxial cables from the control board to the acquisition board(s) must be connected in either right-hand or left-hand pairs of connectors on the control board. Do not mix the connections (i.e., middle two connectors to one board and outside connectors to the other board).

5. If only one acquisition board is installed in the mainframe, connect the round coaxial cables from the control board to the acquisition board in one of the following two ways: J1 of the control board to J3 of the acquisition board and J2 of the control board to J4 of the acquisition board, or J3 of the control board to J3 of the acquisition board and J4 of the control board to J4 of the acquisition board.

6. If two acquisition boards are installed in the mainframe, connect the round coaxial cables from the control boards to the acquisition boards in one of the following two ways: connect one board as defined in step 5, then connect the other board to the remaining two connectors, again, as defined in step 5.

7. Connect the probe pod cable to the middle connector (labeled PROBE) of the acquisition board in the lowest numbered slot. Insert the cable in the cable holder on the mainframe. Label this pod "POD 1" with the label supplied with the pod.

8. If two acquisition boards are installed, connect the remaining pod cable to the middle connector (labeled PROBE) of the remaining acquisition board. Insert the cable in the cable holder on the mainframe. Label the pod "POD 2" with the label supplied with the pod.

9. If other analysis boards are installed in the mainframe and you want communication between the analysis control boards, connect the intermodule bus (IMB) ribbon cable between the connectors on the right-hand side of the boards (as viewed from the rear of the chassis).

NOTE

The connectors on the right side of the acquisition boards are for structural support only. The IMB cable and cable connections to these boards provide the structural support.

REAR-PANEL ADDRESS SWITCHES

The address switches are a set of seven (or five) switches on the mainframe rear panel. Refer to the label located beside the address switches, and set the switches as follows:

1. If you are operating this instrument as part of a system, set switches one through five (or three) to select the system address for your mainframe. Set switches six and seven (or four and five) to select SYSTEM BUS operation.

2. If operating as a stand-alone instrument, switches one through five (or three) are not used. In this case, set switches six and seven (or four and five) to one of the local mass storage (LMS) positions.

MAKING USER COPIES OF FLEXIBLE DISC SOFTWARE

If your timing analyzer was shipped with a set of flexible discs containing your software, you should make a copy of the flexible discs for your use and protect the original set that you received with your system. In your original set is the software to run the timing analyzer, to operate the system functions, to do performance verifications, etc. The procedures in the following paragraphs will show you how to make user flexible discs.

The result of these procedures will be one or two flexible discs having all the software you need for timing analysis. Then you can store the master discs that you got from HP. Use the master set only when you need to make new user flexible discs.

The flexible disc you are going to make will contain the measurement-system routines, the monitor routines, and the timing analysis routines.

When you have performed the procedures described, you will be familiar enough with the copying of discs to make any other disc arrangements you desire. You should also make user flexible discs to contain the performance verification procedures and backup operating system discs.

To make a user set of flexible discs, proceed as follows:

1. Remove two new blank flexible discs from their containers. Label one SYSTEM and the other TIMING. Do not write directly on the discs; this can damage them. Use stick-on labels, if available, or a felt-tip pen. These will become your user flexible discs when the formatting is done.

2. Install the master flexible disc for the operating system (operating system disc #3) in disc drive 0 of your mainframe.

3. Install the new user TIMING disc in disc drive 1.

4. Turn on the LINE power switch. The mainframe will execute the boot-up routines and perform the instrument self test.

5. Press the following keys:

---ETC--- *---ETC---* *floppy* *utilities* **RETURN**

6. The CRT will show an explanation of the floppy utilities routines. A flexible disc must be formatted prior to use. Formatting initializes the disc, preparing it to receive information. Press the following keys:

format **1** **RETURN**

7. When disc 1 formatting is complete, open disc drive 1 and remove your user TIMING disc and install your user SYSTEM disc. Now press the following keys:

format **1 RETURN**

8. When disc 1 formatting is complete, press *end* **RETURN**.

9. Open disc drive 1 and remove your user SYSTEM disc. Install operating system disc #4 in disc drive 1 and press the following keys:

floppy *sys_gen* **RETURN**

10. Remove operating system disc #4 from disc drive 1 and reinstall your user SYSTEM disc in its place. Press the following keys:

copy **FLOPPY__OP__SYS** *from local* **0** *to local* **1 RETURN**

11. When the copy is complete, press the following keys:

copy **MEAS__SYS** *from local* **0** *to local* **1 RETURN**

12. Remove operating system disc #3 from disc drive 0, and install the master timing disc in its place.

13. Remove the user SYSTEM disc from disc drive 1 and install your user TIMING disc. Now press the following keys:

copy **TIMING** *from local* **0** *to local* **1 RETURN**

14. After the copy is complete, press *end* and **RETURN**.

15. Open disc drive 0 and remove the master timing flexible disc. Store all of the original flexible discs from HP in a safe place.

16. Open disc drive 1 and remove your TIMING flexible disc. Your user SYSTEM and TIMING flexible discs have all of the software needed to operate the timing analyzer. Install the SYSTEM disc in disc drive 0 and the TIMING disc in disc drive 1.

CAUTION

The user copy of the TIMING disc cannot be write-protected because temporary configuration files must be written to it.

17. If you are going to operate the timing analyzer along with other analysis modules (such as a state analyzer), you will have to combine the system software and timing analysis software onto one disc. Label it TIMING. Use the above procedure to place all of the software on the TIMING disc. Install your user TIMING disc in disc drive 0 and the flexible disc with software for the other analysis module in disc drive 1.

NOTE

The operating system software and timing analysis software together take most of the disc space. Use one of the following two methods to make more disc space available for storing configurations, etc.

(1). Copy TIMING__SHORT instead of TIMING in step 13. This is the preferred option because it leaves more memory space available on the user TIMING disc. The TIMING__SHORT software is the same as TIMING, except it has no post_process capability.

(2). For the first use of your completed TIMING disc, install it in disc drive 0 and gain access to the timing analyzer (refer to Chapter 3). Now press the *end* softkey. This stores the present timing analyzer configuration in the remaining space on your user TIMING disc. In this way, the system software will automatically save future configurations to the flexible disc in disc drive 1.

PERFORMANCE-VERIFICATION SOFTWARE

Make a working copy of the timing performance verification software as follows:

1. Install operating system disc #3 in disc drive 0.

2. Install a new blank disc (labeled TIMING_PV) in disc drive 1.

3. Press the following keys:

   ---ETC--- *---ETC---* *floppy* *utilities* **RETURN**

4. Format your user TIMING_PV disc by pressing the following keys.

   format **1** **RETURN**

5. When formatting is complete, press *end* and RETURN. Then open disc drive 1 and remove the user TIMING_PV disc. Install operating system disc #4 in disc drive 1 and press the following keys:

   floppy *sys__gen* **RETURN**

6. Remove operating system disc #4 from disc drive 1 and reinstall the user TIMING_PV disc in its place. Press the following keys:

   copy **FLOPPY__OP__SYS** *from* *local* **0** *to* *local* **1** **RETURN**

7. Remove system operating disc #3 from disc drive 0, and install the master timing disc 2 of 2 in its place. Then press the following keys:

   copy **TIMING__PV** *from* *local* **0** *to* *local* **1** **RETURN**

8. When the copy is complete, press *end* and RETURN.

9. Open disc drive 0 and remove the timing master flexible disc 2 of 2. Store it in a safe place.

10. Open disc drive 1, remove your user TIMING_PV disc. Store it for use when making performance verifications of the timing analyzer.

MAKING ADDITIONAL BACKUP FLEXIBLE DISCS

If you have enough spare flexible discs, you should make a second set of your user discs with floppy-utilities copy. In this way, you will always have a master set of flexible discs formatted to your needs. Making future user copies of these discs will be easy because you will only need to copy complete discs instead of making selective copies of several discs.

1. To copy one flexible disc in its entirety, install operating system disc #3 in disc drive 0, and the new blank disc (with appropriate label) in disc drive 1. Then press the following keys:

   ---ETC--- *---ETC---* *floppy* *utilities* **RETURN**

   format **1 RETURN**

2. When formatting is complete, press *end* and **RETURN**.

3. Remove operating system disc #3 from disc drive 0 and install operating system disc #4. Now press the following keys:

   floppy *sys_gen* **RETURN**

4. Remove operating system disc #4 from disc drive 0 and install the disc you want to copy in its place. Then press the following keys:

   copy *all* *from* *local* **0** *to* *local* **1 RETURN**

5. After the copy is complete, press the following keys:

   end **RETURN**

Chapter 3

GETTING STARTED WITH THE TIMING ANALYZER

INTRODUCTION

This chapter provides information pertaining to system software conventions, getting started with timing analysis measurements, becoming familar with the utility softkeys, and configuring the analyzer. This information is provided to help you become familar with the timing analyzer. To do this, you will enter some basic measurement specifications and gather data as a result of these specifications.

SYSTEM SOFTWARE CONVENTIONS

This section contains information concerning the system software as it relates to any of the subsystems installed in a particular mainframe.

ENTERING NUMERIC VALUES

You can enter numbers into a timing analyzer specification in any of the four standard number bases. Place the applicable letter symbol (B, O or Q, D, H) at the end of your number to define its base. Refer to the following examples:

1000B = 1000 binary
1000O or 1000Q = 1000 octal
1000H = 1000 hexadecimal
1000D or 1000 = 1000 decimal

NOTE

> Decimal is assumed if you do not specify the base when you enter a number.

In pattern specifications you can also enter the letter X in combination with the number you enter, in any mode of operation, or you can enter the letter M in the dual threshold mode of operation. The letter X is equal to a "don't care" and the letter M is equal to a "middle". Hexadecimal numbers beginning with a letter, and any numbers to be entered starting with an X or M, must be preceded with a numeric zero. Examples are as follows:

01XB, 0M11B, but not MB (but 0MB)
3FAH, 0FFH, but not F44H (but 0F44H)
07X00, 0X770, but not X710 (but 0X710)
9MH, 0M7H, but not M6H (but 0M6H)

COMMAND FILES

A command file is a source file that contains a sequence of commands as they would appear on the command line if they had been entered manually from the softkeys or keyboard. A command file can be used to create a particular measurement configuration on disc for future use. A command file provides a self-documenting record of a measurement setup and allows easy editing and modification.

FILENAMES

Filenames may consist of from one to nine alphanumeric characters and underscores (_). The first character in the name must be upper-case alpha. All other alpha characters can be upper or lower case.

GETTING STARTED

The procedures in this section are provided to help you become familar with operation of the timing analyzer, and use of the different levels of softkeys. In these procedures, you will connect the timing analyzer to a test signal source. Then you will gain access to the timing analysis function. You will perform timing measurements using the default test setup and you will obtain timing diagrams which show activity in your test signal source. Then you will modify the default test setup and perform special measurements.

SETTING UP THE ANALYZER FOR TEST MEASUREMENTS

1. Determine the points on the circuit under test that you want to probe. If the points are accessible on IC pins, clamp the IC test clip(s) over the IC.

NOTE

> If more than one probe clip is used per IC, connect the probe clip ground jumper wire between the probe clips to prevent spurious pulses from causing erroneous trace displays due to false triggering.

2. Connect the 8 CHANNEL TIMING PROBE cables from the analyzer to appropriate logic signal points on the probe clip or the circuit under test. For purposes of this discussion, MC6800 microprocessor signal names are used. Connections to an MC6800 microprocessor are shown in table 3-1. Connect your timing analyzer to view known parameters in your test system, such as listed in table 3-1, and proceed with the following exercise.

Table 3-1. Test Connections

Signal	MC6800 Pin	Probe Input
CL_0	3	0
CL_1	37	1
R_LW	34	2
A0	9	3
VMA	5	4
D0	33	5
L_RST	40	6
L_IRQ	4	7

3. Connect operating power to the mainframe.

4. In a hard disc based system, boot in the timing analysis software. In a flexible disc based system, install the SYSTEM flexible disc that you created in chapter 2, in drive 0 (left side or top, depending on the mainframe); and the TIMING flexible disc in drive 1. If you placed all of the system and timing software on one flexible disc, install it in drive 0. If other analyzers in addition to timing are installed in the mainframe, the system and timing analysis software must be contained on the flexible disc in drive 0.

5. Turn on the power switch. The associated indicator lamp (on some mainframes) will light.

6. The message "Self-test Completed" will come up on the screen in flexible disc based systems. In hard disc based systems, the units connected to the bus are identified. In either case, the system monitor software is loaded and the monitor level of softkeys is displayed on the bottom of the screen.

7. You may, at this time, wish to assign a user identity code to your activity with the instrument. The software records your userid and assigns any files you may make to your userid. The userid must start with an upper-case alphabetic character and is limited to six characters. After the first letter, the other five characters may be alphanumeric. To assign your userid, press the ---*ETC*--- softkey twice. Press the *userid* softkey, type in the userid you have selected, and press the **RETURN** key. If no userid is selected, the default condition is a blank userid.

8. Turn on the circuit under test.

9. You are now ready to gain access to the timing analyzer.

GAINING ACCESS TO THE TIMING ANALYZER

The first softkey label line at the system monitor level will contain a softkey of either *timing* or *meas__sys*. If a timing analysis module is the only module installed in the mainframe, *timing* will appear. If other modules (state, emulation, additional timing, etc.) are also present in the mainframe, *meas__sys* will appear. Press the *timing* or *meas__sys* softkey (press ---*ETC*--- if necessary to regain the first softkey label line), and press **RETURN**.

If you pressed the *timing* softkey, the system will access the timing analyzer and the default trace specification will appear on the screen.

If you pressed the *meas__sys* softkey, the *timing* softkey will now appear on the screen, along with the names of any other analysis modules in the mainframe. This is the measurement-system level of softkeys. It allows access to any one of the analysis modules installed in the mainframe. Press the *timing* softkey and **RETURN** to gain access to the timing analyzer. The default trace specification will now appear on screen.

NOTE

> If the timing analysis module is not the only analysis module in the mainframe, the *timing* softkey label will appear followed by a number (e.g.; *timing__4*). The number denotes the slot in the mainframe that contains the analyzer control board.

Figure 3-1 shows the syntax for gaining access to the timing analyzer and for switching from one display to another within the timing analyzer. The *show* softkey provides transportation from one specification to another. The *mode* softkey allows you to select the mode of operation from the trace specification or the format specification. The *diagram* softkey allows you to choose 8-channel or 16-channel format of the timing diagram. The *end* softkey allows you to exit the timing analyzer while retaining its present measurement setup and continuing any measurement in progress. After pressing *end*, you can edit, copy, etc., at the system monitor level while the timing analysis measurements are performed and data is retained. You can reenter the timing analyzer later and either continue the trace or read the results of the measurement that was concluded after the *end* softkey was pressed.

You have now gained access to the timing analyzer. Proceed with the instructions in the following paragraphs to perform some basic measurements.

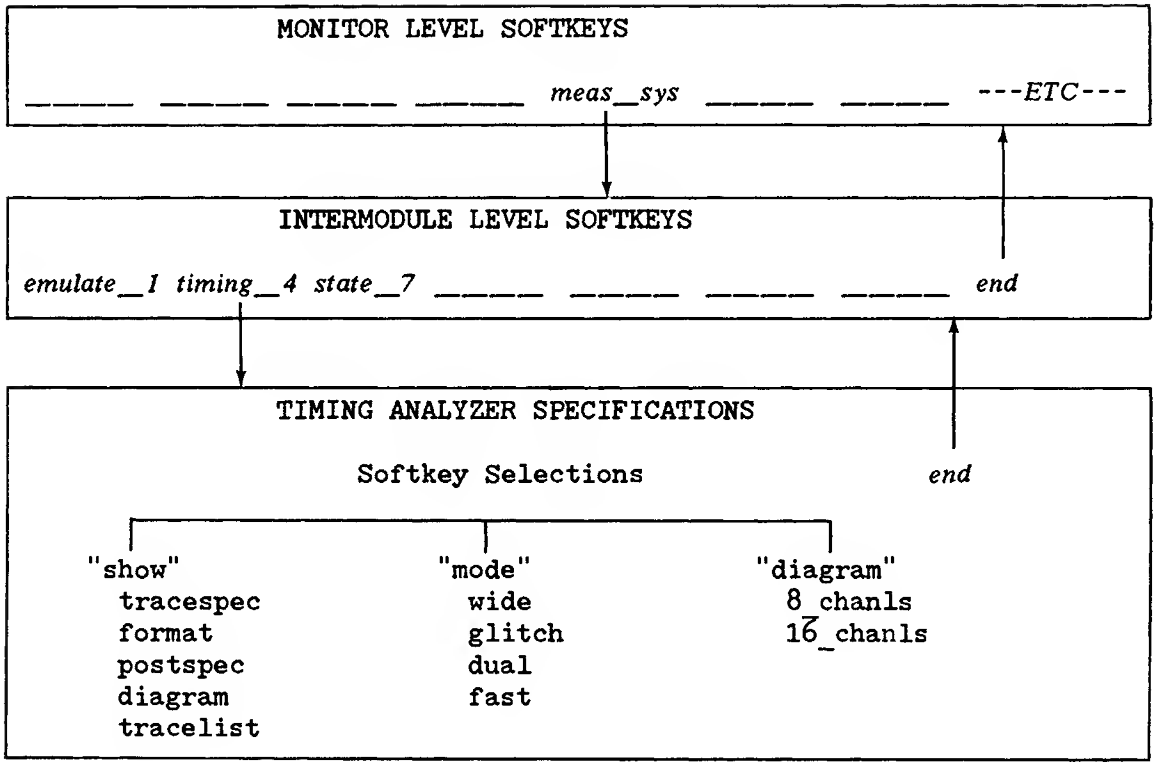

MONITOR LEVEL SOFTKEYS
_____ _____ _____ _____ meas__sys _____ _____ ---ETC---
INTERMODULE LEVEL SOFTKEYS
emulate__1 timing__4 state__7 _____ _____ _____ _____ end
TIMING ANALYZER SPECIFICATIONS
Softkey Selections
end
"show"
tracespec
format
postspec
diagram
tracelist
"mode"
wide
glitch
dual
fast
"diagram"
8_chanls
16_chanls

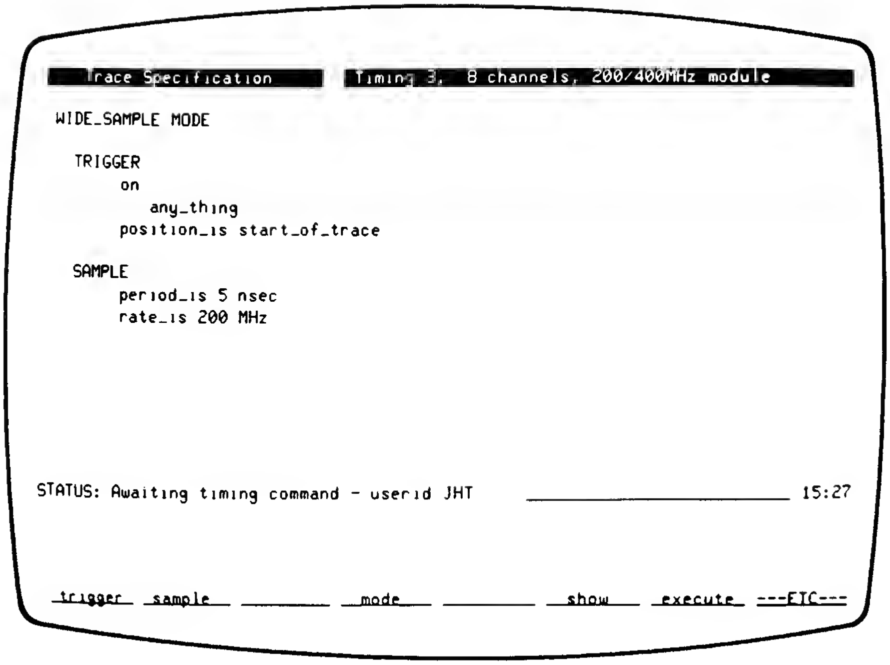
Figure 3-1. Utility Keys Used For Transportation

PERFORMING SOME BASIC (DEFAULT) MEASUREMENTS

1. The default trace specification will be on screen as shown in figure 3-2. Press the *---ETC---* softkey to see the other line of softkey labels available in this specification.

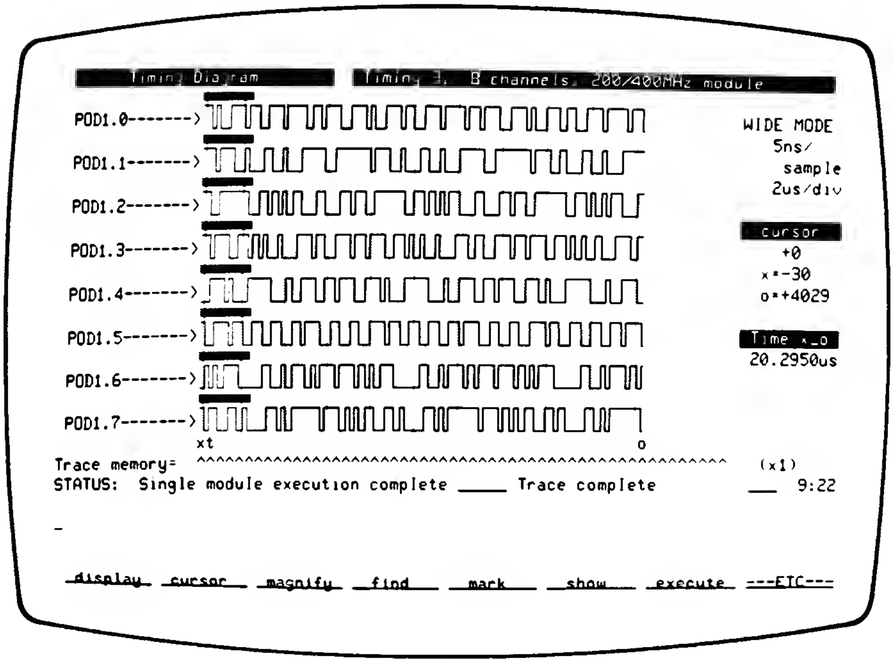
_____ assert default _____ configure copy end ---ETC---

2. Press *---ETC---* again to return to the first line of softkey labels.



```
Trace Specification          Timing 3,  8 channels, 200/400MHz module

WIDE_SAMPLE MODE

  TRIGGER
      on
         any_thing
      position_is start_of_trace

  SAMPLE
      period_is 5 nsec
      rate_is 200 MHz




STATUS: Awaiting timing command - userid JHT    ______________________ 15:27
```

Figure 3-2. Trace Specification Display



```
Timing Diagram          Timing 3,  8 channels, 200/400MHz module
POD1.0------->                                        WIDE MODE
POD1.1------->                                          5ns/
POD1.2------->                                            sample
POD1.3------->                                          2us/div
                                                        cursor
                                                          +0
POD1.4------->                                          x*-30
                                                        o*+4029
POD1.5------->                                          Time x_o
                                                        20.2950us
POD1.6------->
POD1.7------->
              xt                                   o
Trace memory= ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^    (x1)
STATUS:  Single module execution complete ____ Trace complete       ___   9:22

_
```

Figure 3-3. Timing Diagram Default Display

3. Observe that the screen shows WIDE_SAMPLE MODE spelled out in the upper left corner. This is the (default) mode of operation. There are three other modes of operation available: glitch capture, dual threshold, and fast sample. These modes are accessible from both the trace specification and the format specification level softkeys. To access the other modes, press the *mode* softkey, make the desired softkey selection, then press the RETURN key.

4. Return to the wide sample mode by pressing *mode*, *wide*, RETURN. Now press the *execute* softkey, and the RETURN key. This begins a trace under the default trace specification (figure 3-2).

5. When the trace is complete, the timing diagram will be on screen and show a trace of the activity in the probed lines of the circuit under test. See figure 3-3 for the 8-channel default timing diagram. A second line of softkey labels (shown below) is available by pressing the *---ETC---* softkey.

diagram sample indicate <ROLL> configure copy end ---ETC---

6. Press the *execute* and *repeat* softkeys, then the RETURN key. The timing analyzer will begin a series of traces, updating the display after each trace. Press the *halt* softkey, then the RETURN key to stop the trace measurement.

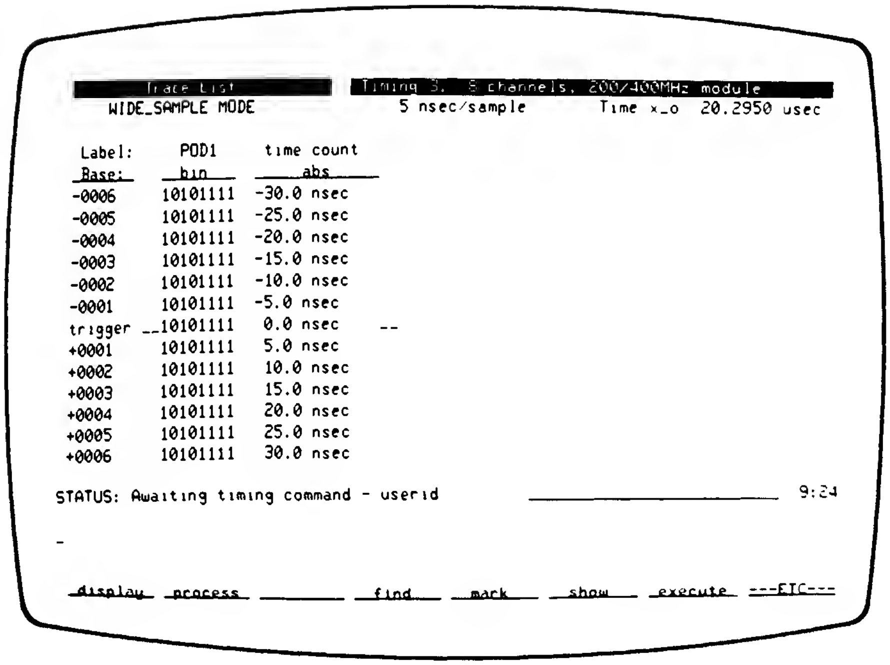
7. Note that the up carats (^) on the "Trace memory =" line extend all the way across the screen, and that (x1) appears at the end of the line. The up carats identify the portion of trace memory being viewed and (x1) defines the horizontal magnification factor. The sample rate and horizontal scale factor are shown under WIDE MODE at the right-hand side of the screen.

8. Press the *magnify* and *x10* softkeys, then the RETURN key. Note the expanded trace and the new scale factor under WIDE MODE on the right side of the screen. Note also that the up-carat portion of the "Trace memory =" line is smaller (denoting that a smaller area of trace memory is being viewed) and that the (x1) at the end of the line has changed to (x10), denoting the higher horizontal magnification factor.

9. Repeat step 8 except press the *x100* softkey instead of the *x10* softkey. Again note the differences in the display.

10. Press the *cursor* softkey. The softkey label will change to inverse video. Now press the --> key. Note that the cursor moves to the right. Note also that the cursor sample number position, shown on the right side of the screen under cursor (in inverse video), changes in number as the cursor moves.

11. Press the *show* and *tracelist* softkeys, then the RETURN key. The default trace list (see figure 3-4) will appear on the screen with POD1 values listed in binary. The time count column shows absolute time between each captured sample and the trigger (on line 0000). The time between mark_x and mark_o is shown at the top, right portion of the screen.



```
   Trace List              Timing 3.  8 channels, 200/400MHz module
    WIDE_SAMPLE MODE                  5 nsec/sample        Time x_o  20.2950 usec

  Label:      POD1      time count
  Base:       bin           abs
 -0006      10101111  -30.0 nsec
 -0005      10101111  -25.0 nsec
 -0004      10101111  -20.0 nsec
 -0003      10101111  -15.0 nsec
 -0002      10101111  -10.0 nsec
 -0001      10101111  -5.0 nsec
 trigger --10101111   0.0 nsec     --
 +0001      10101111   5.0 nsec
 +0002      10101111   10.0 nsec
 +0003      10101111   15.0 nsec
 +0004      10101111   20.0 nsec
 +0005      10101111   25.0 nsec
 +0006      10101111   30.0 nsec

 STATUS: Awaiting timing command - userid        ____________________    9:24

 -

 display  process  ________  find  mark  show  execute  ---ETC---
```

Figure 3-4. Trace List Specification Default Display

12. Press the *find* and *mark_x* softkeys, then the RETURN key. Note that the display window moves to the location of mark_x (line -0030) in the trace memory.

13. Press the *find* and *mark_o* softkeys, then the RETURN key. Note that the display window moves to the location of mark_o (line +4029) in the trace memory.

14. Press the *mark*, *x*, and *on_sample* softkeys, then type in 1000 and press the RETURN key. Note the new measurement of time between mark_x and mark_o in the upper, right-hand side of the display.

NOTE

> "mark_x" and "mark_o" will always be assigned to samples in memory. You can assign "mark_x" to any sample in the trace memory and assign "mark_o" to any other sample. The measurement between the two marked samples will be shown on the display. Mark_x and mark_o have default assignments of "begin" and "end" of trace. If a "halt" occurs, "mark_x" and "mark_o" will move to their default assignments.

15. Using the **ROLL UP**, **ROLL DOWN**, **NEXT PAGE**, **PREV PAGE**, and the up and down arrow keys, note the movement of the cursor line and the trace list on the screen.

PERFORMING SOME SPECIFIC MEASUREMENTS

You have completed some basic procedures using the default capabilities of the timing analyzer. Proceed with the following instructions to set up a measurement configuration more suited to a specific requirement.

NOTE

> The following procedures will be performed using the signal names listed in table 3-1. If you are obtaining your input signals from a source other than an MC6800, the labels you assign will not be accurate identifiers. They will still serve to demonstrate the use of labels in the timing analyzer.

1. Press the *show* and *format* softkeys, then the **RETURN** key. The format specification will appear on screen. You will now assign labels to your probed points for ease of reference.

2. Press the *define* softkey. The softkey label line will contain the default labels and the *<LABEL>* softkey prompt. Type in **CL_0**. Labels must start with a capital letter and can then contain up to eight alphanumeric characters; underscores (_) are permissible.

3. After you have typed in your label, the softkey label line will contain the *pod_1_bit* softkey. If you have a sixteen-channel analyzer, the softkey label line will also contain the *pod_2_bit* softkey. Press the *pod_1_bit* softkey.

NOTE

> For purposes of this discussion it is assumed that the eight-channel option is installed. The sixteen-channel analyzer allows you additional features which, if not obvious from the eight-channel discussion, will be explained.

4. Type **0** in answer to the *<BIT#>* prompt softkey, then press the **RETURN** key. Your new label will appear as the last label in the specification. The command line on screen will appear as follows:

define CL_0 pod_1_bit 0

5. Define labels for the rest of the inputs as shown in figure 3-5. Use the procedure in steps 2 through 4 or use the **-->**, **<--**, **TAB**, **SHIFT TAB**, **INSERT CHAR**, or **DELETE CHAR** keys to modify the command line to define each new label. Press **RETURN** to enter each new label definition into the format specification.

6. After you are finished assigning the labels, the format specification will look like figure 3-5. A second line of softkey labels (shown below) is available by pressing the *---ETC---* softkey.

delete rename default ______ configure copy end ---ETC---

7. Now that your labels are assigned, it is time to set up a trigger. Press the *show* and *tracespec* softkeys, then the **RETURN** key. The trace specification (see figure 3-2) will appear on the screen.

8. Set up a trigger as follows:

   a. Press the *trigger*, *on*, *entering*, *R__LW*, *=* softkeys. Type in **1** from the keyboard. Press the *and*, *---ETC---*, *VMA*, *=* softkeys, and again type in **1** from the keyboard. Press the *and*, *CL__1*, *=* softkeys, and again type in **1** from the keyboard. The command line on screen will appear as follows:

   trigger on entering R__LW = 1 and VMA = 1 and CL__1 = 1

   b. Press the **RETURN** key. The trigger you have just set up will now appear on screen under *TRIGGER*.

   c. Press the *trigger*, *position*, *center* softkeys, and then the **RETURN** key. Note that the new trigger position also appears on screen under *TRIGGER*.

9. Press the *show diagram* softkeys, then the **RETURN** key.

10. Set up the screen display as follows:

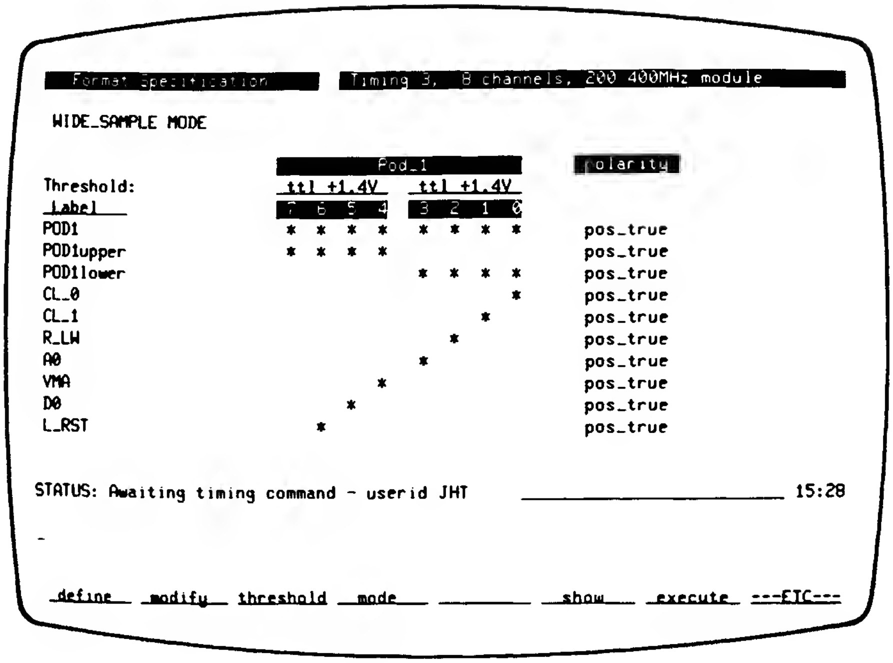
    a. Press the *display ---ETC---* softkeys. The labels you assigned when you made up the format specification will be present on the softkey label line.

    b. Press the softkeys in the order shown below to display the input signals identified by your labels.

    display CL__0 then CL__1 then R__LW then A0 then

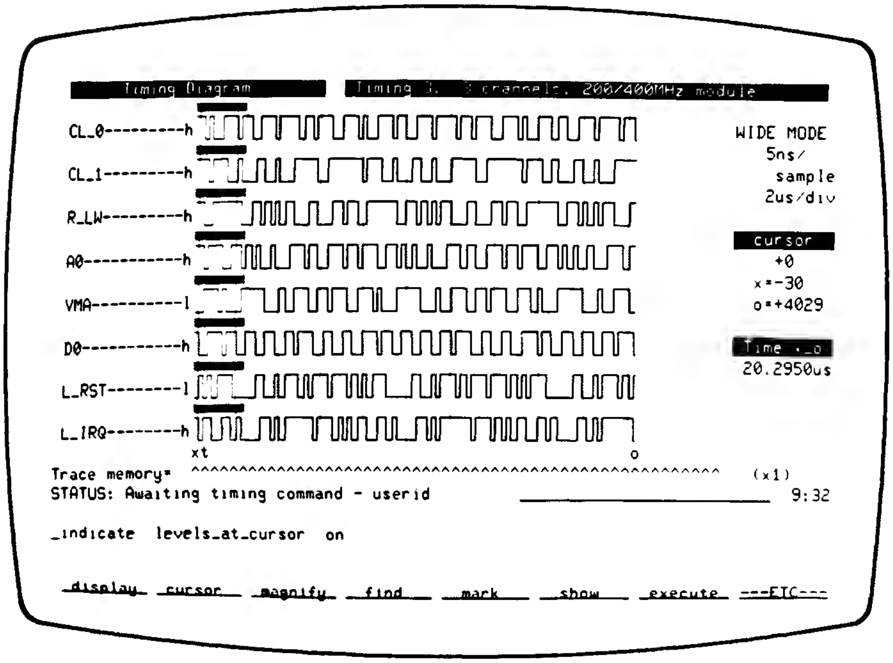
    VMA then D0 then L__RST then L__IRQ **RETURN**

    c. The display will show the signals identified by the labels you assigned, instead of by the default labels (POD1.0 through POD1.7).



```
 Format Specification          Timing 3,  8 channels, 200 400MHz module

 WIDE_SAMPLE MODE

                           Pod_1                 polarity
Threshold:           ttl +1.4V  ttl +1.4V
 Label               7  6  5  4  3  2  1  0
POD1                 *  *  *  *  *  *  *  *    pos_true
POD1upper            *  *  *  *                pos_true
POD1lower                        *  *  *  *    pos_true
CL_0                                      *    pos_true
CL_1                                   *       pos_true
R_LW                                *          pos_true
A0                               *             pos_true
VMA                          *                 pos_true
D0                        *                    pos_true
L_RST                  *                       pos_true

STATUS: Awaiting timing command - userid JHT   ______________ 15:28

-

 define   modify  threshold  mode   ________  show   execute  ---ETC---
```

Figure 3-5. 8-Channel Format Specification



```
   Timing Diagram          Timing 3,  8 channels, 200/400MHz module

CL_0---------h                                       WIDE MODE
                                                        5ns/
CL_1---------h                                           sample
                                                        2us/div
R_LW---------h
                                                       cursor
A0-----------h                                           +0
                                                        x=-30
VMA----------l                                          o=+4029

D0-----------h                                        Time x_o
                                                      20.2950us
L_RST--------l

L_IRQ--------h
               xt                              o
Trace memory= ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^   (x1)
STATUS: Awaiting timing command - userid  ______________________  9:32

_indicate  levels_at_cursor  on

 display  cursor   magnify   find    mark    show    execute  ---ETC---
```

Figure 3-6. Timing Diagram Display

11. Press the *magnify x1* softkeys, and the **RETURN** key. This allows you to start with a display that views the contents of all of the trace memory. Now press the *execute* softkey, then the **RETURN** key.

12. To obtain trigger recognition, the timing analyzer must first find the trigger condition not-true. Then it will recognize trigger on the first occurrence of the trigger being true. If trigger recognition occurs, the timing analyzer will complete a trace. If the trigger you specified is not found in the incoming data, press the *halt* softkey and **RETURN**. The screen will show a timing diagram similar to that in figure 3-6, except that the levels (h or l) will not be present at the end of each label. To show the levels, press the ---*ETC*---, *indicate*, *levels*, *on* softkeys, then the **RETURN** key. The level indicators will now be present. The 'h' and 'l' indicate whether the associated trace is high or low at the position of the cursor.

13. Press the *cursor* softkey. The *cursor* softkey label will change to inverse video. This signifies that the --> and <-- keys can move the cursor line on the display in the direction of the arrow.

14. Using the --> and <-- keys, move the cursor to a positive-going edge on one of the traces. Press the *magnify x100* softkeys, then the **RETURN** key. Again align the cursor with the positive-going edge on the selected trace.

NOTE

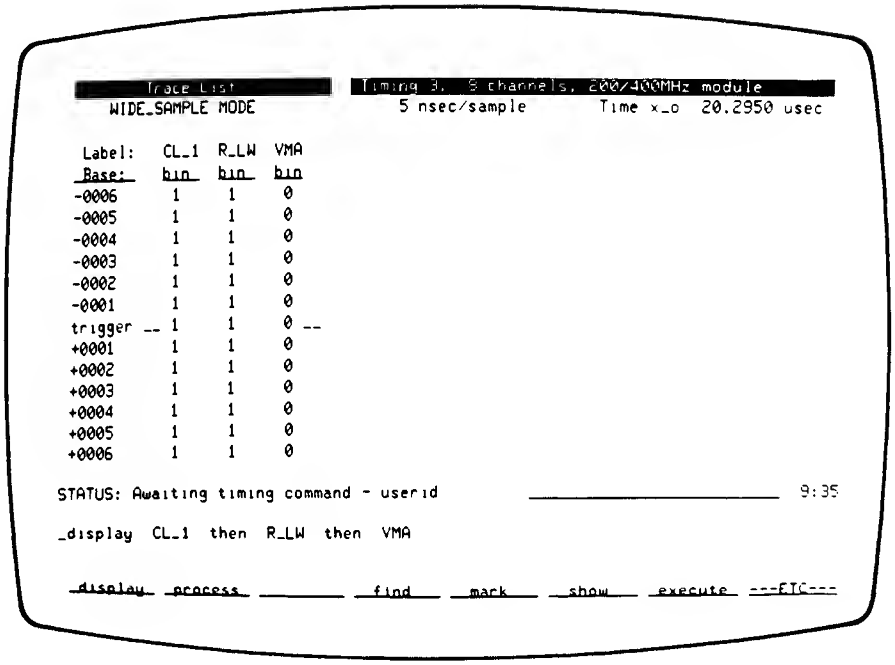
> If the selected trace does not show a positive-going edge, reduce the magnification factor to x10 to find it. After it is found and lined up with the cursor, then select the x100 magnification factor to obtain the most accurate cursor placement.

15. Press the *mark x* softkeys, then the **RETURN** key. The mark x will move to the point where the cursor is positioned. The sample number where mark_x is assigned is shown at the middle, right side of the screen under *cursor*.

16. Press the *magnify x1* softkeys, then the **RETURN** key. This places the complete trace memory on screen. Using the --> and <-- keys, move the cursor to a positive-going edge on another trace. Press the *magnify x100* softkeys, then the **RETURN** key. Again align the cursor with the positive-going edge of the selected trace. Note that you may have to use the x10 magnification factor, as in step 14, to find the positive-going edge of the trace.

17. Press the *mark o* softkeys, then the **RETURN** key. The mark o will move to the point where the cursor is positioned. The sample number where mark_o is assigned is shown at the middle, right side of the screen under *cursor*.

18. The time between mark x (positive-going edge on the first trace) and mark o (positive-going edge on the second trace) is shown at the right-hand side of the screen under *timex_o*.



```
      Trace List            Timing 3.  9 channels, 200/400MHz module
   WIDE_SAMPLE MODE              5 nsec/sample        Time x_o  20.2950 usec

 Label:   CL_1  R_LW  VMA
 Base:    bin   bin   bin
-0006      1     1     0
-0005      1     1     0
-0004      1     1     0
-0003      1     1     0
-0002      1     1     0
-0001      1     1     0
trigger -- 1     1     0 --
+0001      1     1     0
+0002      1     1     0
+0003      1     1     0
+0004      1     1     0
+0005      1     1     0
+0006      1     1     0

STATUS: Awaiting timing command - userid     ____________________     9:35

_display  CL_1  then  R_LW  then  VMA

 display  process  ________  find  mark  show  execute  ---ETC---
```

Figure 3-7. Trace List Display Diagram

19. Press the *show* and *tracelist* softkeys, then the **RETURN** key. The trace list will now appear on screen.

20. Set up the trace list to view only the traces that were defined as part of the trigger conditions in the trace specification by pressing the following keys:

display *---ETC---* *CL__1* *then* *---ETC---* *R__LW* *then*

---ETC--- *VMA* **RETURN**

21. Press the *find* and *trigger* softkeys and **RETURN**. If the timing analyzer found the trigger and completed its trace, the display on the screen will be like that shown in figure 3-7. If not, the display will show lines -0013 through -0001 on screen because the trigger (line 0000) was never found. A second line of softkey labels (shown below) is available by pressing the *---ETC---* softkey.

______ *sample* *indicate* *<ROLL>* *configure* *copy* *end* *---ETC---*

22. If the trigger was found, then the trigger is in the center of the list on line 0000 and the trigger conditions set up in the trace specification are present at the trigger (e.g., R_LW = 1, VMA = 1, and CL_1 = 1). Note also that the time between mark_x and mark_o is shown on the top, right side of the screen, and that it agrees with the reading on the timing diagram taken in step 18.

23. Press the *configure save_in* softkeys, then press the A and **RETURN** keys on the keyboard. With this entry, the timing analyzer stores your entire measurement setup in a trace file which it will name A under the presently defined userid.

24. Press the *show format* softkeys, then press the **RETURN** key. The format specification will appear on screen.

25. Press the *---ETC---*, *default*, *all_specs* softkeys, then the **RETURN** key. These entries default the entire timing analyzer setup.

26. Press *---ETC---* and enter the following command line: *configure load_from* A, and press **RETURN**. This reloads the measurement specification that you stored in the file named A.

27. Press the *show tracespec* softkeys, then the **RETURN** key. This calls the trace specification to the screen. Notice that your trace specification has been returned to the setup that you stored in the file named A.

This completes an introduction to the timing analyzer. You have connected the instrument to a test circuit and run a few simple tests. You have modified the measurement specifications to obtain specific information, and then you have stored and recovered your measurement setup using memory. The remainder of this chapter provides detailed information that will help you obtain greater benefit from the utility offered by the timing analyzer.

CONFIGURING THE ANALYZER

There are four measurement configurations which you can have the analyzer load automatically when you first activate it: (1) the default setup, (2) the measurement setup you used to perform the last tests, (3) any measurement setup that was stored in a trace file, and (4) any measurement system command file.

GETTING THE DEFAULT CONFIGURATION

The default configuration for the timing analyzer is as follows:

Format Specification (Eight channel)

```
Labels: POD1, POD1upper, POD1lower
```

Format Specification (Sixteen Channel)

```
Labels: POD1, POD1upper, POD1lower
        POD2, POD2upper, POD2lower
```

Trace Specification

```
Mode: WIDE_SAMPLE
Trigger: on anything
         position is start of trace
Sample: period is 5 nsec
        rate is 200 MHz
```

Timing Diagram Display Specification (Eight Channel)

```
Display: POD1, POD1upper, POD1lower
         default_bits_in pod_1
```

Timing Diagram Display Specification (Sixteen Channel)

```
Display: POD1, POD1upper, POD1lower
         POD2, POD2upper, POD2lower
         default_bits_in pod_1
         default_bits_in pod_2
```

Trace List Display Specification (Eight Channel)

```
Display: POD1, POD1upper, POD1lower
         default_bits_in pod_1
Base: Binary
Count Time: Absolute
```

Trace List Display Specification (Sixteen Channel)

```
Display: POD1, POD1upper, POD1lower
         POD2, POD2upper, POD2lower
         default_bits_in pod_1
         default_bits_in pod_2
Base: Binary
Time Count: Absolute
```

When you have gained access to the timing analyzer by pressing the *timing* softkey, the default measurement setup is automatically entered in the analyzer. At this time, you can press the *execute* softkey and run a test.

Since the default measurement specification probably will not make the exact test that you want to make, you can modify the setup. Use the softkeys in the format specification and trace specification to set up any unique measurement desired.

GETTING THE MEASUREMENT CONFIGURATION USED LAST

When you press the *end* softkey, the analyzer will store the present measurement specification. It will even continue to run a test according to this specification if you have a test in progress. To return the timing analyzer to that setup, press the *timing* (or *meas_sys*) and *continue* softkeys, then the **RETURN** key. If you had a measurement in process when you pressed the *end* softkey, it may still be running when you reenter, if it was not already completed.

GETTING A MEASUREMENT CONFIGURATION FROM A TRACE FILE

The timing analyzer can store complete measurement configurations in trace files so that you can keep a library of test setups on hand for your measurement needs. The following procedure shows how to store and recover these measurement configurations.

1. Set up any desired measurement configuration in your timing analyzer.

2. Press the *configure* and *save_in* softkeys, then press the A keyboard key. This will set up the analyzer to save its present measurement configuration in a trace file named A under the current USERID. Now press the *with_data* softkey. This instructs the timing analyzer to additionally save all of the data in the trace memory when it saves the measurement configuration in file A. Press **RETURN**.

3. Now you can change the setup any way you like. Your original measurement configuration will still be saved exactly as you stored it in file A.

4. Press *end* to return to the monitor level of softkeys.

5. Press the *timing* softkey, then the A and **RETURN** keyboard keys. You will gain access to the timing analyzer and it will automatically search the disc and load the configuration you stored in file A. It will also load the trace memory with the data you stored in file A. Use this method when you are entering the timing analyzer and you want to load a special measurement configuration at the same time.

6. If you are operating the timing analyzer and want to reload the configuration in file A without first ending out, press *configure load_from* and A RETURN. This will cause the timing analyzer to purge the present measurement setup and load the configuration you saved in file A (along with data if you saved data in that file).

7. When you want to save a measurement configuration for future use, you can save time and memory space by not including *with_data* in your command. This will set up the timing analyzer to save just the measurement configuration in your file.

8. You can use the above procedures to save as many different measurement configurations as you like, each under a different name in memory. Then you can recall and use these configurations for later testing. You can also modify the content of any of these configurations once you have reloaded them into the timing analyzer.

GETTING A MEASUREMENT CONFIGURATION FROM A COMMAND FILE

You can write a command file to activate the timing analyzer and perform an automatic measurement. You can write your command file into a larger command file that controls activities in several instruments. To write a command file for the timing analyzer, obtain the monitor level of softkeys and enter the Editor mode, as follows:

edit into CMDFILE (use any command file name)

Write your command file with the sequence of command lines that you would use to create the desired setup from the monitor level. When writing your command file, be sure to use the names that appear on the command line for each entry, not the names of the softkeys (use *show format_specification* not *show format*). The parameter-passing feature of command files allows you to include the slot number of the timing control board in the command. An example command file is shown below:

```
PARMS &SLOTNUMBER

 measurement_system

 timing _&SLOTNUMBER

 show format_specification

 define ONE pod_1_bit 0

 show trace_specification

 trigger on entering ONE = 1

 show timing_diagram

 display ONE then POD1

 execute
```

When you run the example command file, the display will prompt you for the slot number of the timing analyzer control board using the following message:

Define parameter &SLOTNUMBER:

Type the number of the slot where the timing analyzer control board is installed. Then the timing analyzer will execute the remainder of the command file.

If the timing analyzer control board is in slot number 3, for example, you can make the example program run without operator action by replacing the first three lines with the following:

measurement__system

timing__3

If the timing analyzer is the only measurement unit present in the mainframe, replace the first three lines in the example command file with the single line:

timing

When you have completed your command file, press the *end* softkey and **RETURN**.

To run the command file from the system monitor level, type in the command file name, and press **RETURN**. For example:

CMDFILE 3 (not *run* **CMDFILE 3**) **RETURN**

If the slot number of the timing analyzer control board is needed, and your command file does not allow for that parameter, the timing analyzer will display the message:

Syntax error

Chapter 4

UTILITY KEYS

INTRODUCTION

The utility keys consist of the utility softkeys, utility keyboard keys, and the prompt softkeys. These keys are described in detail in the following paragraphs.

UTILITY SOFTKEYS

The utility softkeys are common to all five specifications (trace, format, post_process, timing diagram, and trace list), except as noted in the following paragraphs. The utility softkeys allow you to *show* any one of the five displays for viewing or modifying in any one of the four modes of operation. You can *execute* or *halt* a measurement, *save* or *load* a particular configuration, *default* a particular specification (or all specifications), *copy* any or all of the displays, and *end* the analysis session without losing your current specifications or stopping a measurement in progress. The utility commands are described in detail in the following paragraphs.

MODE SOFTKEY

The *mode* softkey is available only in the trace and format specifications. The *mode* softkey allows you to choose from four different operating modes, depending upon the particular application desired. To access a particular mode when in another mode, press the *mode* softkey, the softkey with the name of the next mode desired, and then the **RETURN** key. The modes and their functions are described in the following paragraphs.

1. **Wide Sample Mode.** The wide sample mode is the most versatile and, therefore, the most commonly used mode. This mode is the only mode that allows use of all eight input channels of the timing probe. Use it for all timing measurements not requiring the specialized functions of the other modes. The wide sample mode is the default mode of the timing analyzer.

2. **Glitch Capture Mode.** The glitch capture mode is used to detect and display the occurrence of multiple data transitions between adjacent data samples, i.e.; within one sample period. If more than one transition is detected between samples, this information is stored in a glitch memory and displayed on screen. A glitch is displayed using a special symbol: a broken vertical bar. In x100 magnification, a glitch is shown by five adjacent broken vertical bars. In x1 and x10 magnifications, a single broken vertical bar is used to indicate a glitch as well as to indicate the occurrence of multiple data transitions too close together to be displayed at the selected horizontal magnification.

3. **Dual Threshold Mode.** The dual threshold mode sets up a middle voltage band to identify times when the voltage is between separately presetable upper and lower thresholds, established in the format specification. This mode can be used to identify times when the voltage is between the threshold limits of the TTL V_{IH} min and the V_{IL} max (as would be caused, for example, by excessive fanout or bus contention). This condition is indicated by a middle level in the trace.

4. **Fast Sample Mode.** The fast sample mode is used to obtain very high time resolution of sampled data. The sample rate in this mode is fixed at 400 MHz (2.5-nsec period), giving the trace at least twice the resolution of the other modes. It also has twice the memory depth of the other modes.

SHOW SOFTKEY

The *show* softkey allows you to select the trace specification, format specification, post_process specification, timing diagram, or the trace list for viewing and/or manipulating or modifying the current measurement setup on screen.

SAMPLE SOFTKEY

The *sample* softkey is only used in the trace specification, trace list, and timing diagram displays. It is used to set either the sample rate or the sample period of the analyzer, except during the fast sample mode where they are preset to 400 MHz and 2.5 nsec, respectively. In the wide sample and dual threshold modes, the sample rate is adjustable from 200 MHz to 2 Hz and the sample period is adjustable from 5 nsec to 500 msec. In the glitch capture mode, the sample rate is adjustable from 100 MHz to 2 Hz and the sample period is adjustable from 10 nsec to 500 msec. To set the sample rate or sample period, press the *sample* softkey. The softkey label line will show the *period* and *rate* softkeys. Press the one you wish to set, then type in the time period or frequency rate you wish to set, and press the **RETURN** key.

NOTE

The sample rate is selectable as a multiple of 1, 2, or 4. If the selected unit is not a true multiple, the number is rounded upward; i.e., if 12 MHz is selected, 20 MHz will appear in the SAMPLE specification on screen and will be the value recognized by the analyzer.

The sample period is selectable as a multiple of 10, 25, or 50. If the selected unit is not a true multiple, the number is rounded downward; i.e., if 23 nsec is selected, 10 nsec will appear in the SAMPLE specification on screen and will be the value recognized by the analyzer. The analyzer will only accept integers, however, if you specify a period of 2500 nsec, "period_is 2.5 usec" will appear on screen.

EXECUTE SOFTKEY

The *execute* softkey enables the analyzer to initiate a measurement based on the parameters set up in the trace specification. Pressing the *execute* softkey, then the **RETURN** key causes the analyzer to load the measurement setup. When this is accomplished, the STATUS line will flash "Waiting for trigger" and the analyzer will begin filling trace memory. When the trigger condition set up in the trace specification is satisfied, trace memory will continue to be filled until the trigger position relative to the trace is satisfied. When this requirement is also satisfied, "Trace Complete" will appear on screen and a new timing diagram will be displayed.

If the trigger specification cannot be satisfied, no trace will appear and the STATUS line will blink "Waiting for trigger".

The *repeat* softkey, when used with the *execute* softkey, causes repetitive executions of the trace measurement. The results of each measurement are displayed on screen. If the *repeat* softkey is not used, the *execute* softkey must be pressed each time to start a new measurement.

HALT SOFTKEY

When the *execute* softkey is pressed, *execute* on the softkey label line is replaced with *halt*. The label *halt* will remain until the measurement is completed and results are displayed on screen, or until the *halt* softkey is pressed. Pressing the *halt* softkey causes the measurement in process to be stopped. If no trigger was found, all data accumulated in trace memory will be displayed. If a trigger was found, all data in trace memory accumulated after the trigger was found will be displayed.

CONFIGURE SOFTKEY

The *configure* softkey is used to either save a configuration you have set up, or to load a configuration from a file which you previously saved. To save a configuration, press the *configure* and *save_in* softkeys and type in the file name you will use to identify the configuration. If you want to save the data in the trace memory along with the instrument configuration so you can use it for comparison measurements, press the *with_data* softkey. If you want the file to be write-protected, press the *protect* softkey. Now press the **RETURN** key. To load a configuration you have previously saved, press the *configure* and *load_from* softkeys, type in the name of the file you want to load, and then press the **RETURN** key. If you saved your configuration 'with-data', your trace memory will automatically be loaded with that data during this operation.

NOTE

> If you have used the write-protect option, and want to purge a file, you must return to the system monitor level to accomplish the purge. Press the *end* softkey, then the **RETURN** key to get to the monitor level. Purge the unwanted file, then return to the timing analyzer by using the *continue* option or by entering a command file name along with the *timing* or *meas__sys* command. If desired, you can also use the default setup by pressing the *timing* or *meas__sys* softkey to enter the analysis function.

When the logic timing analyzer is operating in a stand-alone mode using flexible-disc based software, *hpib* is available as a configuration source or destination. In this case, the *configure* softkey performs a setup function. If you *configure save__to* or *configure load__from* an *hpib* controller, your command will set up the logic timing analyzer to perform either as a "talker" or a "listener". The HP-IB controller (connected to the HP-IB port) will determine when data transfer begins and ends, and how the configuration files will be identified and stored.

DEFAULT SOFTKEY

The *default* softkey is used to reset all, or any one, of the analyzer specifications to their default conditions. This is accomplished by pressing *default* and *all__specs*, or by pressing *default* and the appropriate specification softkey, then the **RETURN** key.

NOTE

> The *default* softkey is present only in the trace, format, and post_process specifications. You must be in one of those specifications to default any or all of the specifications.

COPY SOFTKEY

The *copy* softkey allows you to copy the trace specification, format specification, post_process specification, or all three to a file or to a printer. It also allows you to copy the timing diagram on screen, or all or selected parts of the trace list to a file or to a printer. To copy the trace specification to a printer, press the *copy*, *tracespec*, *to*, *printer* softkeys, and then the **RETURN** key.

END SOFTKEY

The *end* softkey is used for transportation from the present software level to the next higher level. When you press *end*, the system will exit the timing analyzer and return to either the measurement_system display or monitor level display. When you leave the analyzer by use of the *end* softkey, your measurement configuration will be stored on the disc. If your timing analyzer has a test in progress, it will continue the test without interruption after you press *end*. Your data will be collected automatically. You can use the monitor level softkey functions without disturbing the measurement you ended from as long as you do not press the *opt_test* softkey at the monitor level. You can reenter the timing analysis module at any time and observe the results of your test by pressing the *timing* (or *meas_sys*) and *continue* softkeys and the RETURN key.

NOTE

> If you do not include the *continue* statement in your command, your present measurement configuration will be purged along with all data collected.

---ETC--- SOFTKEY

The *---ETC---* softkey, when pressed, causes the next line of softkey labels to be displayed on screen. If the last line is currently on display, pressing this softkey will return the first line of softkey labels to the screen.

UTILITY KEYBOARD KEYS

RECALL - The **RECALL** key will cause the analysis module to return the preceding command line to the screen. The analysis module has a command line memory which the **RECALL** key accesses. Each time you press **RECALL**, the analyzer steps one execution further back into its memory of command lines.

TAB - The **TAB** key is used to move the cursor rapidly through the measurement specification on screen. This key is useful when you are making modifications to long measurement specifications. By pressing **TAB**, you step the cursor from entry to entry forward through the specification. By pressing **SHIFT** and then **TAB**, you step the cursor backwards through the specification.

INSERT/DELETE - The **INSERT** and **DELETE** keys are used to edit the content of the command line. The **INSERT** key will open a space before the present position of the cursor so that you can enter additional requirements in the measurement specification. The remainder of the specification will automatically shift to the right with each entry you make. The **INSERT** key function will remain in effect until it is pressed again or until any other utility key is pressed (except **<--**, **-->**, or **CAPS LOCK**). The **DELETE** key is used to elimimate characters from the specification without losing the entire specification. When you press the **DELETE** key, the entry directly over the cursor will be eliminated and the remainder of the specification will shift left. Holding the **DELETE** key down will cause multiple character deletions as characters are shifted left, over the cursor position.

CONFIGURATION FILE UTILITY

There is a *wait* utility key that does not show up on any display, but is available for use in command files. If you type in *wait*, the logic timing analyzer will offer three parameters: *meas_comp*, *<TIMER>*, and *<RETURN>*. These three parameters indicate the following types of wait functions:

a. *meas_comp*. To enter this parameter in your command file, type in *wait measurement_complete*. Use this parameter after each *execute* command to ensure that the measurement is completed in the logic timing analyzer before the next command arrives from the command file. If you do not include the "wait measurement_complete" statement, the next command in the command file will be introduced as soon as the measurement execution begins. This parameter is expecially useful when preparing a command file that requires repetitive executions of a measurement, such as *halt_repetitive_execution when_time_x_o greater_than 50 usec*. It will ensure that the measurement will be executed as many times as necessary to capture the specified data.

b. *<TIMER>*. To use this parameter in your command file, type in *wait 47 seconds* or any other time period you select. This command is useful when preparing a command file that calls up a series of selected displays. You may want each display to remain on screen for a period of time before proceeding to the next display.

c. *<RETURN>*. To use this parameter in your command file, type in *wait*. This command causes the command file to wait until the operator presses any keyboard key. By using this command, you can present displays on screen and leave them there until the operator decides to call for the next step in the command file.

NOTE

> You can overrive the effect of any of the three delay parameters by pressing any keyboard key during the *wait* period. By pressing any keyboard key, you will introduce the next command from the command file.

PROMPT SOFTKEYS

Any softkey name enclosed in angle-brackets "<>" is a prompt softkey. If you press a prompt softkey, the STATUS line on the display will explain the meaning of the prompt.

Chapter 5

FORMAT SPECIFICATION

INTRODUCTION

This chapter describes the softkeys that are specific to the format specification menu. These softkeys, along with the utility softkeys described in chapter 4 are used to format the timing analyzer. The softkeys in this menu are described in detail on the following pages. The format specification display in some of the options will vary, depending on whether the instrument is set up for 8-channel or 16-channel operation. Eight-channel operation is discussed in this chapter. Where there are basic operational differences between 8-channel and 16-channel operation, both are discussed. The 8-channel default format specification is shown in figure 5-1. A second line of softkey labels (shown below) is available by pressing the *---ETC--* softkey.

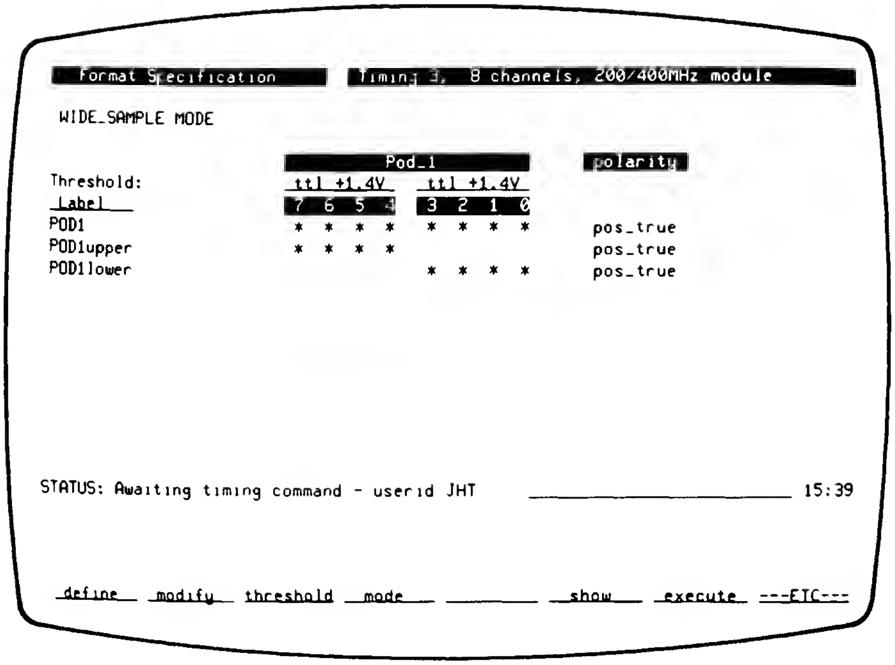
delete rename default ______ *configure copy end ---ETC---*

FORMAT SPECIFICATION SOFTKEYS

The format specification softkeys consist of the utility and prompt softkeys, and the *threshold*, *define*, *modify*, *delete*, and *rename* softkeys. The utility softkeys in the format specification are the *mode*, *show*, *execute*, *halt*, *configure*, *default*, *copy*, *end*, and *---ETC---* softkeys. The utility and prompt softkeys are discussed in chapter 4. The format specification-specific softkeys are described in the following paragraphs. The *mode* utility softkey is discussed briefly to introduce the format specification for each mode.

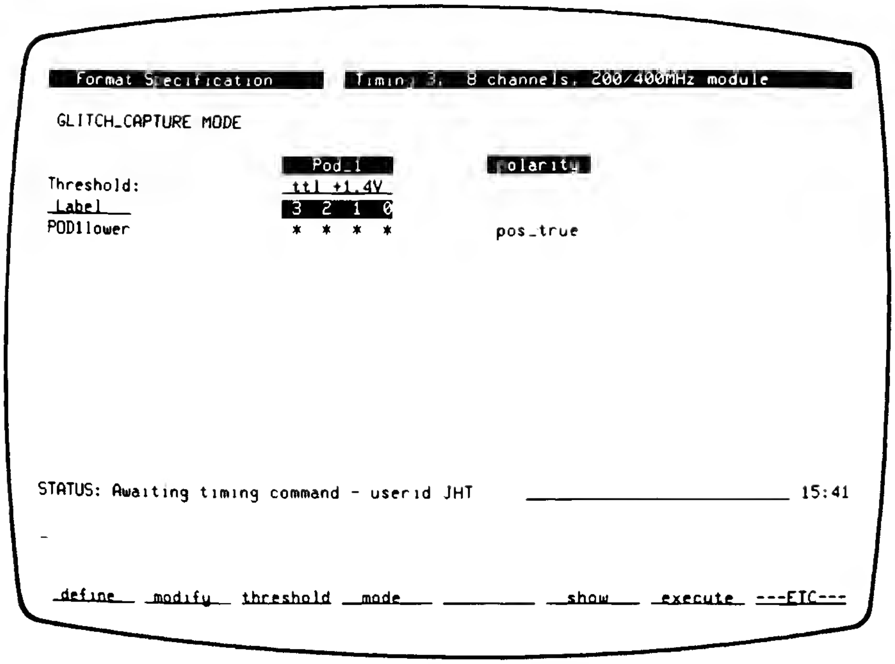
MODE SOFTKEY

The *mode* softkey allows you to choose one of the four operating modes: wide sample mode, glitch capture mode, dual threshold mode, and fast sample mode. The default format specifications for these modes are shown in figures 5-1 through 5-4, respectively. Each mode is discussed in detail in chapter 4 under "Utility Softkeys".



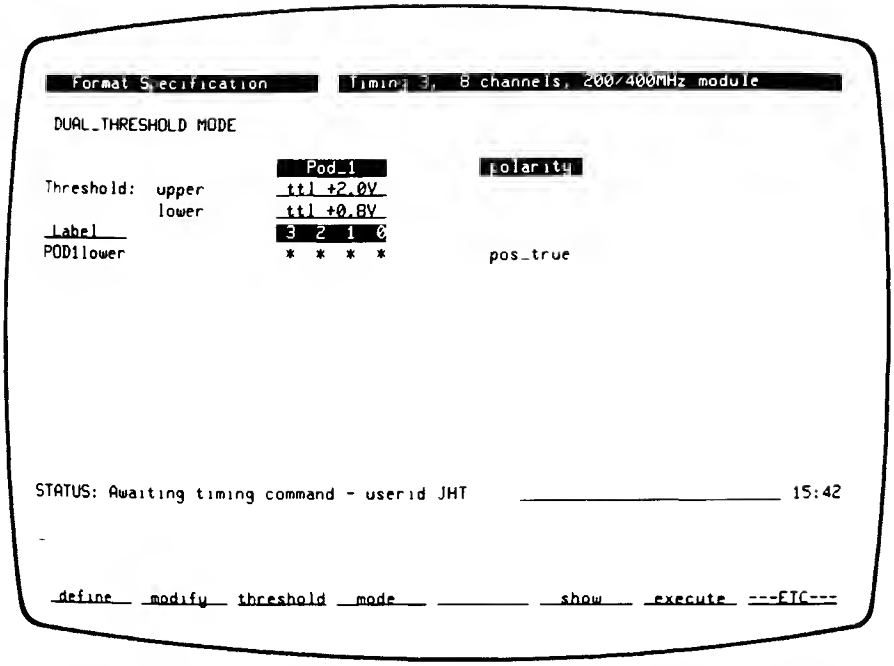
```
Format Specification          Timing 3, 8 channels, 200/400MHz module

WIDE_SAMPLE MODE

                                Pod_1                 polarity
Threshold:               ttl +1.4V    ttl +1.4V
Label                    7  6  5  4   3  2  1  0
POD1                     *  *  *  *   *  *  *  *      pos_true
POD1upper                *  *  *  *                   pos_true
POD1lower                             *  *  *  *      pos_true

STATUS: Awaiting timing command - userid JHT     ________________  15:39
```

Figure 5-1. 8-Channel Wide__Sample Default Format Specification



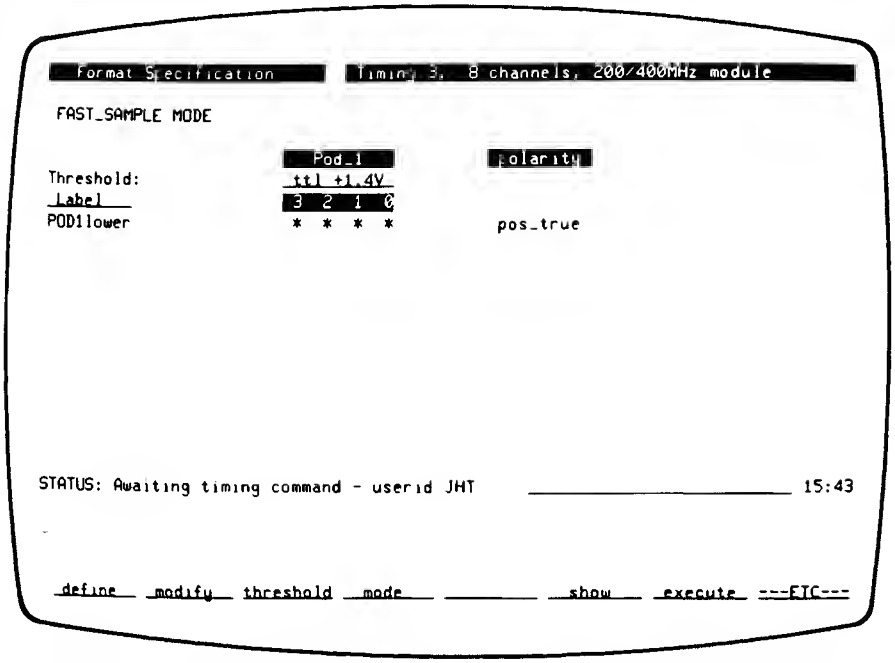
```
Format Specification          Timing 3, 8 channels, 200/400MHz module

GLITCH_CAPTURE MODE

                           Pod_1         polarity
Threshold:               ttl +1.4V
Label                    3  2  1  0
POD1lower                *  *  *  *      pos_true

STATUS: Awaiting timing command - userid JHT     ________________  15:41
```

Figure 5-2. 8-Channel Glitch__Capture Default Format Specification



```
 Format Specification          Timing 3,  8 channels, 200/400MHz module

 DUAL_THRESHOLD MODE

                                 Pod_1            polarity
Threshold:  upper              ttl +2.0V
            lower              ttl +0.8V
 Label                          3  2  1  0
POD1lower                       *  *  *  *          pos_true




STATUS: Awaiting timing command - userid JHT   ______________________  15:42

 -

 define   modify   threshold   mode   ________   show   execute   ---ETC---
```

Figure 5-3. 8-Channel Dual__Threshold Default Format Specification



```
 Format Specification          Timing 3,  8 channels, 200/400MHz module

 FAST_SAMPLE MODE

                                 Pod_1            polarity
Threshold:                     ttl +1.4V
 Label                          3  2  1  0
POD1lower                       *  *  *  *          pos_true




STATUS: Awaiting timing command - userid JHT   ______________________  15:43

 -

 define   modify   threshold   mode   ________   show   execute   ---ETC---
```

Figure 5-4. 8-Channel Fast__Sample Default Format Specification

THRESHOLD SOFTKEY

The *threshold* softkey allows setting the threshold levels of the eight bits in the probe pod. The threshold levels available in the wide, glitch, and fast modes are ttl (+1.4V), ecl (-1.3V), or any voltage between -10.0 and +10.0 volts (in steps of 0.1 volt). The threshold levels in the dual threshold mode are ttl (upper = +2.0V, lower = +0.8V), ecl (upper = -1.1V, lower = -1.5V), or the same voltage range as in the other modes; the lower threshold must be more negative than the upper threshold. All eight bits of the pod may be set to one of the above values, or the upper four bits may be set to one value and the lower four bits may be set to another.

DEFINE SOFTKEY

The *define* softkey allows you to define labels to identify the activity found on each of the probe bits. Each probe pod has eight bits (0 through 7) and each bit has a separate column in the format specification (see figure 5-1). An asterisk in a column indicates that the bit will be represented by the associated label in the label column. Labels may be defined to identify single bits or sets of bits. Up to 31 separate labels may be defined in one format specification. The default labels (shown in figure 5-1) are POD1, POD1upper, and POD1lower for eight-channel operation. The default logic polarity of labels is positive-true. Softkey selection allows you to change the definition and/or logic polarity of any label.

If only one or two points are being probed during a measurement, the points are easy to remember. When more points are probed, it is more difficult to associate displayed traces with the probed points. By assigning relevant labels, the association of input bits with probed points is established for the measurement. By using labels, it is easier to document measurement results, and easier to use the data for review after time has elapsed.

Labels are easy to define. For example, suppose you have bit 1 connected to address line A10 of a microprocessor under test. To assign the label, A10, to bit 1, press the *define* softkey, type in **A10** from the keyboard, press the *pod_1_bit* softkey, type in **1** from the keyboard, and press the RETURN key. Note that the display now shows A10 under the Label column and an asterisk under Pod_1 bit 1, denoting that the label A10 is assigned to identify bit 1 of Pod 1. The label A10 will also appear as a softkey in the trace specification where it can be used for triggering definitions and in the timing diagram where it can be used to format the display on screen.

Labels can be defined to identify multiple inputs. This is especially convenient for setting up a trigger in the trace specification. If you want to trigger a trace when all eight bits of pod 1 are in selected conditions, press the *define* softkey, type in the label name you want to use, press the *pod_1_bit* softkey, type in **0** in response to the *<BIT#>* prompt, press the *thru* softkey, type in **7** in response to the *<BIT#>* prompt, then press the **RETURN** key. Your label will appear on screen and asterisks will be present under each bit number (0 through 7) on the label line. A trigger can now be set up in the trace specification based on the condition of all eight bits of the pod. For example; the trigger specification setup could be:

trigger on entering <LABEL> = 11XX0010B **RETURN**

Disjointed (noncontiguous) labels can also be defined. However, when using these labels (in a trigger for instance) it must be noted that the bit number within a label is determined by its location as shown in the format specification. For example; define the label **DATA** as follows:

define DATA pod__1__bit 1 and pod__1__bit 3 and pod__1__bit 5 and pod__1__bit 7 **RETURN**

The label DATA will now appear as the last label on screen. It is composed of four bits. The bit numbers are determined by counting from right to left, starting with zero. Hence, the label DATA is composed of bit 0 (pod 1, bit 1), bit 1 (pod 1, bit 3), bit 2 (pod 1, bit 5), and bit 3 (pod 1, bit 7). The trigger specification will only accept this format. The following is an example trigger specification using the DATA label defined above:

trigger on entering DATA .0 = 1 **RETURN**

Because bit 0 in the DATA label is pod_1_bit_1, the above trigger specification is the same as:

trigger on entering POD1lower .1 = 1 **RETURN**

As stated previously, the signal polarity may be specified as *logic__polarity positive__true* (the default condition) or *negative__true*. This can be accomplished when the label is defined or by modifying the label once it has been defined. Note that the signal polarity selected applies to the trigger definition (in the trace specification) and to the trace list. The timing diagram, however, will always show positive voltage up and negative voltage down on the display.

MODIFY SOFTKEY

The *modify* softkey is used to modify any of the default or user-defined labels. Press the *modify* softkey and note that the labels listed on the screen under the Label column are also shown on the softkey label line. Now press the *POD1* softkey and the **RETURN** key. The command line will appear as follows:

__define POD1 pod__1__bit 0 width 8 logic__polarity positive__true

Using the -->, <--, TAB, or **SHIFT TAB** key, move the cursor under the *8* and type in a **6** from the keyboard. Press the **RETURN** key and note that the POD1 label asterisks now extend under bits 0 through 5 only. Any of the terms beyond *define* on the command line can be changed in this manner, either by softkey or keyboard entries.

DELETE SOFTKEY

The *delete* softkey can delete any label in the label column, whether default or user-defined, provided it is not used in any trace specification or post_process specification in any mode. All labels not used in the trace specification or post_process specification can be deleted at one time by pressing the *delete* and *all_label* softkeys, then the **RETURN** key. Labels not used in the trace specification can be removed one at a time by pressing *delete* and the softkey for the individual label in place of the *all_label* softkey.

RENAME SOFTKEY

The *rename* softkey can be used to rename any label appearing on the softkey label line. Press *rename* and the softkey which has the label you wish to rename. Next, press the *to* softkey. The prompt *<LABEL>* will appear, prompting you to type in the new label. After the new label is typed in, press the **RETURN** key. The screen will show the new label. If the renamed label was used in any of the other specifications, it will automatically be renamed in those places.

Chapter 6

TRACE SPECIFICATION

INTRODUCTION

This chapter describes the menu of softkeys that are specific to the trace specification. These softkeys, along with the utility softkeys described in chapter 4 are used to set up the trace specification for the timing analyzer. The softkeys that are included in this menu are described in detail on the following pages. The trace specification display and some of the options will vary, depending on whether the instrument is set up for 8-channel or 16-channel operation. Eight-channel operation is discussed in this chapter unless there are basic operational differences. In those cases, both operational concepts are discussed. The 8-channel default trace specification is shown in figure 6-1. A second line of softkey labels (shown below) is available by pressing the ---*ETC*-- softkey.

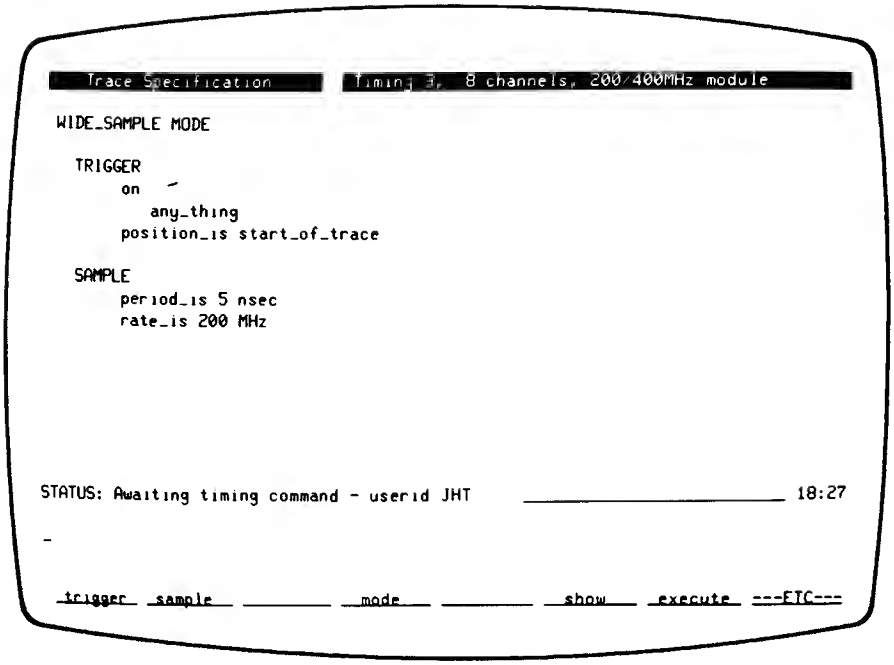
______ *assert* *default* ______ *configure* *copy* *end* ---*ETC*---

TRACE SPECIFICATION SOFTKEYS

The trace specification softkeys consist of the utility and prompt softkeys, and the *trigger*, and *assert* softkeys. The utility softkeys in the trace specification consist of the *mode*, *show*, *sample*, *execute*, *halt*, *configure*, *default*, *copy*, *end*, and ---*ETC*--- softkeys. These softkeys are discussed in detail in chapter 4. The prompt softkeys are also discussed in chapter 4. The trace specification-specific softkeys are described in the following paragraphs. The *mode* utility softkey is discussed briefly to introduce the trace specification for each mode.

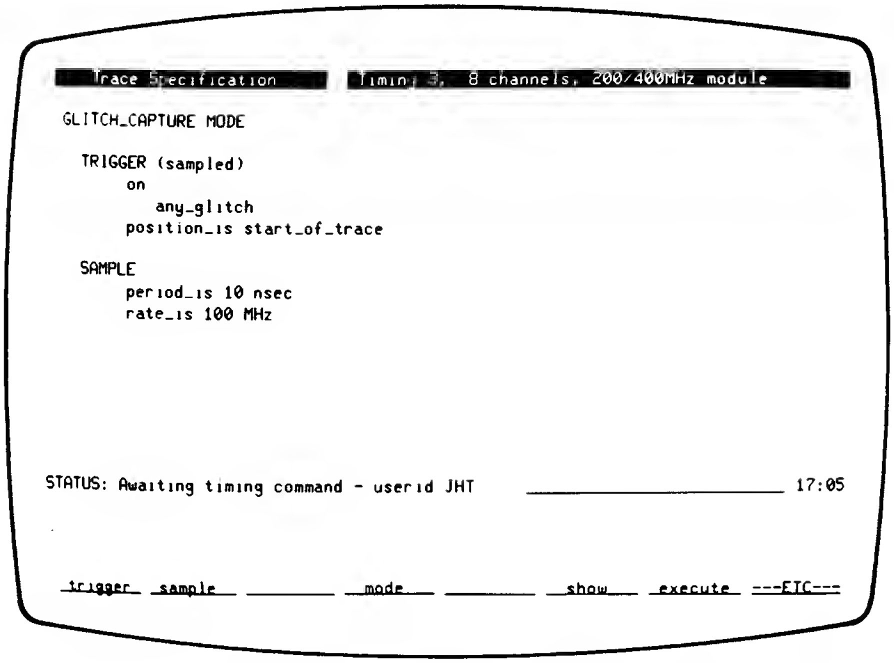
MODE SOFTKEY

The *mode* softkey allows you to choose from four different operating modes as follows: wide sample mode, glitch capture mode, dual threshold mode, and fast sample mode. The default format specifications for these modes are shown in figures 6-1 through 6-4, respectively. The different modes are discussed in detail in chapter 4 under "Utility Softkeys".



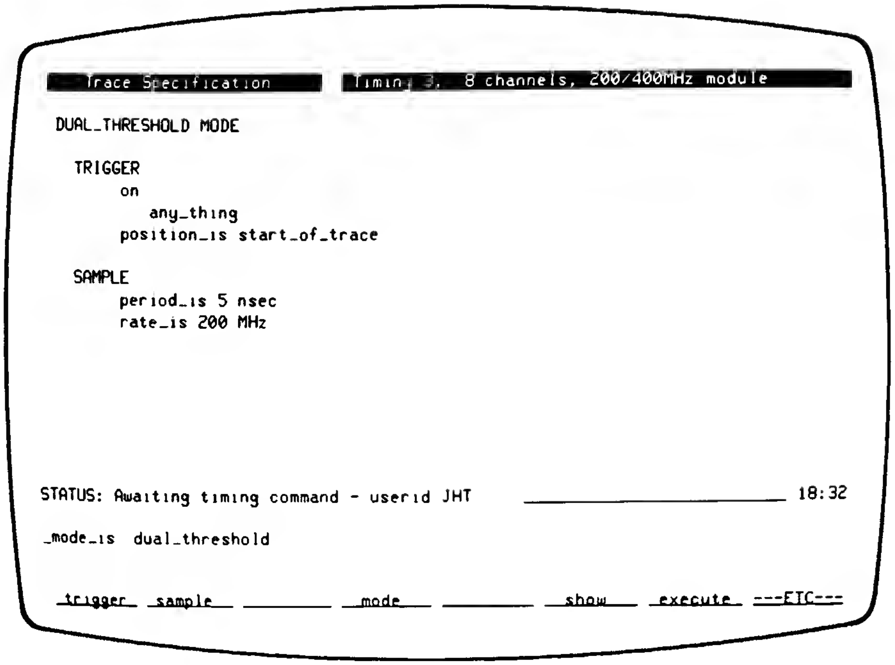
```
 Trace Specification      Timing 3,  8 channels, 200/400MHz module
WIDE_SAMPLE MODE

  TRIGGER
      on
         any_thing
      position_is start_of_trace

  SAMPLE
      period_is 5 nsec
      rate_is 200 MHz




STATUS: Awaiting timing command - userid JHT    ____________________ 18:27

-

 trigger  sample  ________  mode  ________  show  execute  ---ETC---
```

Figure 6-1. 8-Channel Wide__Sample Default Trace Specification



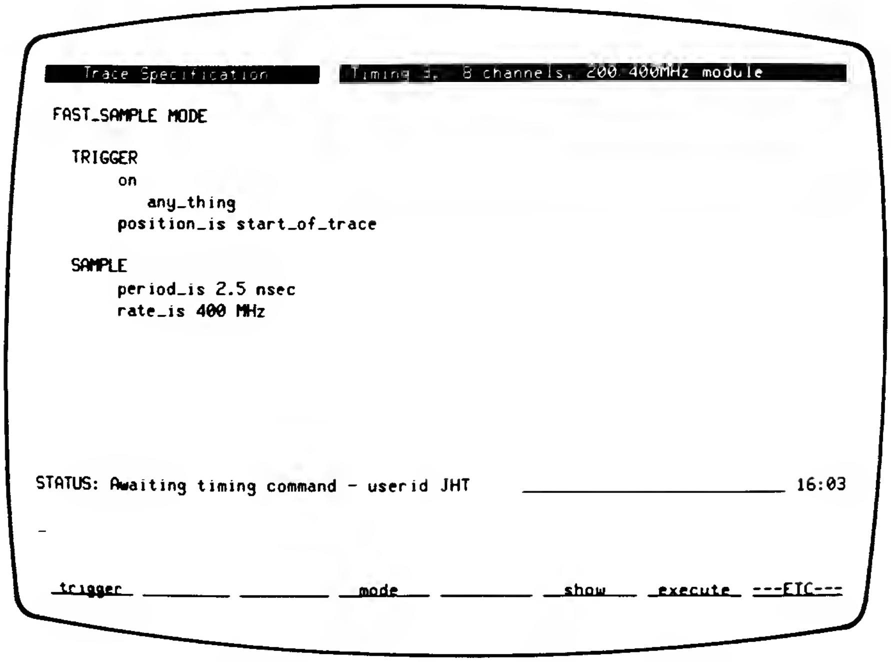
```
 Trace Specification      Timing 3,  8 channels, 200/400MHz module
GLITCH_CAPTURE MODE

  TRIGGER (sampled)
      on
         any_glitch
      position_is start_of_trace

  SAMPLE
      period_is 10 nsec
      rate_is 100 MHz




STATUS: Awaiting timing command - userid JHT    ____________________ 17:05


 trigger  sample  ________  mode  ________  show  execute  ---ETC---
```

Figure 6-2. 8-Channel Glitch__Capture Default Trace Specification



```
Trace Specification          Timing 3,  8 channels, 200/400MHz module

DUAL_THRESHOLD MODE

  TRIGGER
      on
         any_thing
      position_is start_of_trace

  SAMPLE
      period_is 5 nsec
      rate_is 200 MHz





STATUS: Awaiting timing command - userid JHT   _______________________ 18:32

_mode_is  dual_threshold

 _trigger_ _sample_ ________ _mode_ ________ _show_ _execute_ ---ETC---
```

Figure 6-3. 8-Channel Dual__Threshold Default Trace Specification



```
Trace Specification          Timing 3,  8 channels, 200 400MHz module

FAST_SAMPLE MODE

  TRIGGER
      on
         any_thing
      position_is start_of_trace

  SAMPLE
      period_is 2.5 nsec
      rate_is 400 MHz





STATUS: Awaiting timing command - userid JHT   _______________________ 16:03

_

 _trigger_ ________ ________ _mode_ ________ _show_ _execute_ ---ETC---
```

Figure 6-4. 8-Channel Fast__Sample Default Trace Specification

TRIGGER SOFTKEY (8-CHANNEL OPERATION)

The *trigger* softkey allows you to define the parameters that will trigger the trace and to define the trigger position in the trace memory. Refer to chapter 3 for a definition of the numeric values that can be entered to define the trigger parameters. General rules, options, and examples are given below.

General Rules. The general rules listed below must be followed when constructing a trigger specification.

1. The value of any probe bit may be defined only once in a trigger specification, i.e.; different labels which include the same pod bits (as defined in the format specification) cannot be used in their entirety to set up trigger parameters. For example, the default format specification calls out POD1, POD1lower, and POD1upper as labels, therefore, a trigger specification which calls for "POD1upper = 1111B and POD1.7 = 1" will not be accepted because the same bit (pod 1, bit 7) is defined using both labels. Note that this is not acceptable even though both labels define the bit to be the same value. However, parts of labels (as defined by the label bit number) may be used together in the same trigger specification. For example, "POD1upper = 1111B and POD1.0 = 1" is acceptable as a trigger specification. Note that in the second example POD1.0 is not part of POD1upper.

2. When the trigger specification includes a label that has more than one bit, the numeric value of the label should define all bits in the label. For example, if the label DATA is defined in the format specification to include bits 1, 3, 4, and 5 of POD1, and you want to specify a trigger on the DATA label, then *trigger on entering DATA = 1110B* is a valid specification. If the value you enter for "DATA" has too many characters, the leading bits (MSB's) will truncated. If the value has too few characters, leading zeros will be assumed. Examples are as follows:

```
DATA = 11100B  (11100B will be entered as 1100B)

DATA = 10B     (10B will be entered as 0010B)
```

Be sure to consider the effect of this convention when using hexidecimal, decimal, or octal values. For example, POD1 = 1H is the same as POD1 = 00000001B.

3. When <> is chosen as part of the trigger definition, the *and* option is no longer available; the label used in the definition must cover all bits to be defined.

4. When the first trigger parameter is defined as "= <PATTERN>", the <> option is no longer available. Conversely, when the first trigger parameter is defined as <> <PATTERN>, the = option is no longer available.

5. The *and* parameter on the softkey label line is optional. The **RETURN** key can be pressed after entering the first parameter definition or anytime thereafter that a new parameter is entered and the *<RETURN>* softkey prompt appears on the softkey label line. Whenever the *<RETURN>* softkey prompt appears on the softkey label line, it signifies that the command syntax is valid to the cursor.

Options and Examples. The *trigger on* parameters that can be selected during 8-channel operation are shown below, along with some examples for clarification. Where more than one example is given, the examples are separated by an "or" in the middle of the page.

1. **any_thing:**

trigger on any_thing **RETURN**

2. **entering or leaving** a label or group of labels whose value(s) are = or <> to a pattern assigned by the user:

trigger on entering POD1upper = 1101B and POD1lower = 0001B **RETURN**

or

trigger on entering POD1upper <> 1101B **RETURN**

3. **entering or leaving** a bit or group of bits whose value(s) are = or <> to a pattern assigned by the user:

trigger on leaving POD1lower .0 <> 1 **RETURN**

or

trigger on leaving POD1lower .0 = 1 and POD1lower .1 = 0 and POD1lower .2 = 1 and POD1lower .3 = 0 and POD1upper .0 = 1 and POD1upper .1 = 0 and POD1upper .2 = 1 and POD1upper .3 = 1 **RETURN**

This type of trigger specification is especially useful when triggering a trace on an occurrence of known parameters (e.g., specific microprocessor operations). In this case, labels can be assigned to the different bits for ease of recognition. Refer to chapter 5 for a description of label assignments.

Note that the above trigger specification is equivalent to:
trigger on leaving POD1lower = 0101B and POD1upper = 1101B
or
trigger on leaving POD1 = 11010101B
or
trigger on leaving POD1 = 0D5H

4. greater than or less than a selectable time period of a label or group of labels (or a bit or group of bits) being = or <> to a pattern (assigned by the user):

trigger on greater_than 5 nsec_of POD1 = 0AFH **RETURN**

or

trigger on greater_than 5 nsec_of POD1lower .0 <> 1

RETURN

5. The trace point can be delayed by a period of time or number of IMB positive clock edges after the trigger, established in steps 1 through 4 above. This is accomplished by typing in a value from the keyboard in response to the *<DELAY>* softkey prompt; selecting the *ns_after*, *us_after*, *ms_after*, *s_after*, or the *clk_after* softkey, and continuing with the "trigger on" process defined in 1 through 4. For example:

trigger 10 ns_after entering POD1upper .0 = 1 **RETURN**

The time delay may be specified in increments of twice the sample clock period, from 2 to 32 million periods.

6. The trigger can be modified by pressing the *trigger* and *modify* softkeys, and the **RETURN** key. The present trigger specification will appear on the command line. Use the -->, <--, **TAB**, or **SHIFT** and **TAB** keys to move the cursor on the command line. The trigger specification can be modified using the softkeys, the **INSERT CHAR** and **DELETE CHAR** keys, or the keyboard keys, as appropriate.

7. The trigger position can be defined as the start, center, or end of the trace. It can also be defined as a selectable percentage (in multiples of 5%) before the end of the trace or after the start of the trace. Examples of these conditions are as follows:

trigger position_is start_of_trace **RETURN**

or

trigger position_is 5 percent_after_start_of_trace **RETURN**

TRIGGER SOFTKEY (16-CHANNEL OPERATION)

The *trigger* softkey in 16-channel operation provides all the trigger capabilities of an 8-channel subsystem for either pod 1 or pod 2 (refer to the 8-channel triggering discussion). The 16-channel subsystem also provides additional capabilities for triggering on defined conditions between pods. General rules, and additional options and examples are given below.

General Rules. The general rules to follow when constructing a trigger specification are the same as those given for the 8-channel subsystem, with the following additions:

1. When <> is chosen as an option for a label defined in both pods, the *and* and *or* options are no longer available.

2. The *and* and *or* parameters on the softkey label line are optional. The **RETURN** key can be pressed after entering the first parameter definition or anytime thereafter that a new parameter is entered and the *<RETURN>* softkey prompt appears on the softkey label line. Whenever the *<RETURN>* softkey prompt appears on the softkey label line, it signifies that the command syntax is valid to the cursor.

Options and Examples. The additional "trigger on" parameters that are available for selection during 16-channel operation are shown below, along with some examples for clarification.

1. "ANDing" conditions between two pods:

trigger on entering POD1 = 00H and POD2 = 0FFH

RETURN

2. "ORing " a condition in one pod with a condition in the other pod:

trigger on entering POD1 = 00H or POD2 = 0FFH

RETURN

3. Sequencing a condition in one pod followed by a condition in the other pod:

trigger on greater_than 100 nsec_of POD1 = 0FFH

followed_by entering POD2 = 0AAH **RETURN**

4. The simultaneous occurrence of a condition in one pod combined with a time-qualified condition in the other pod:

trigger on entering POD1 = 07H when__greater__than 50 nsec__of

POD2 = 0AH **RETURN**

or (in the glitch_capture mode)

trigger on any__glitch on POD1lower when__greater__than

1 usec__of POD2lower = 0AH **RETURN**

The glitch_capture mode implies a sampled_data trigger condition, where the state of <LABEL> = <PATTERN> can change only at the internal sample clock edge. Because of this, the "when_greater_than" time duration must be at least twice as long as the sample period to ensure it is sampled more than one time.

ASSERT SOFTKEY

The *assert* softkey is used to select assertion of the trigger signal to BNC port number 4 on the instrument rear panel. If asserted, the trigger will be present on the BNC as long as the trigger specification shown on the screen is satisfied. To assert the trigger at BNC port 4, press the *assert*, *bnc__4*, *on__triggr* softkeys, and the **RETURN** key. The assert specification will be displayed on screen. To remove the trigger from BNC 4, press the *assert*, *bnc__4*, *off* softkeys, and the **RETURN** key. The trigger will no longer be on BNC 4, and the specification will be removed from the screen. Terminate the trigger signal from BNC 4 in a 50-ohm input to obtain a good signal waveshape.

Chapter 7

TIMING DIAGRAM DISPLAY SPECIFICATION

INTRODUCTION

This chapter describes the menu of softkeys that are specific to the timing diagram display specification. These softkeys, along with the utility softkeys described in chapter 4 are used to set up the timing diagram display specification for the timing analyzer. The softkeys that are included in this menu are described in detail on the following pages. Syntax diagrams for the more complex softkeys are shown in Appendix A.

The timing diagram display specification and some of the options will vary, depending on whether the instrument is set up for 8-channel or 16-channel operation. Eight-channel operation is discussed in this chapter except where there are basic operational differences. In those cases, both operational concepts are discussed. The 8-channel default timing diagram display specification is shown in figure 7-1. A second line of softkey labels (shown below) is available by pressing the *---ETC---* softkey.

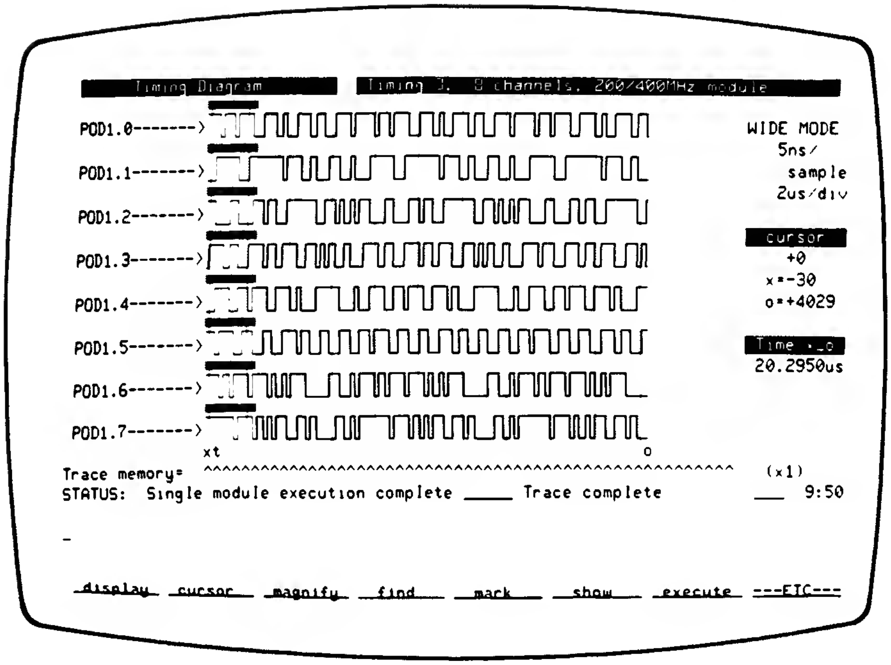
diagram sample indicate <ROLL> configure copy end ---ETC---

The arrows on the diagram at the ends of the label lines point to the middle of each trace when the timing analyzer is displaying 8 channels, and to the top of each trace when displaying 16 channels.

TIMING DIAGRAM DISPLAY SPECIFICATION SOFTKEYS

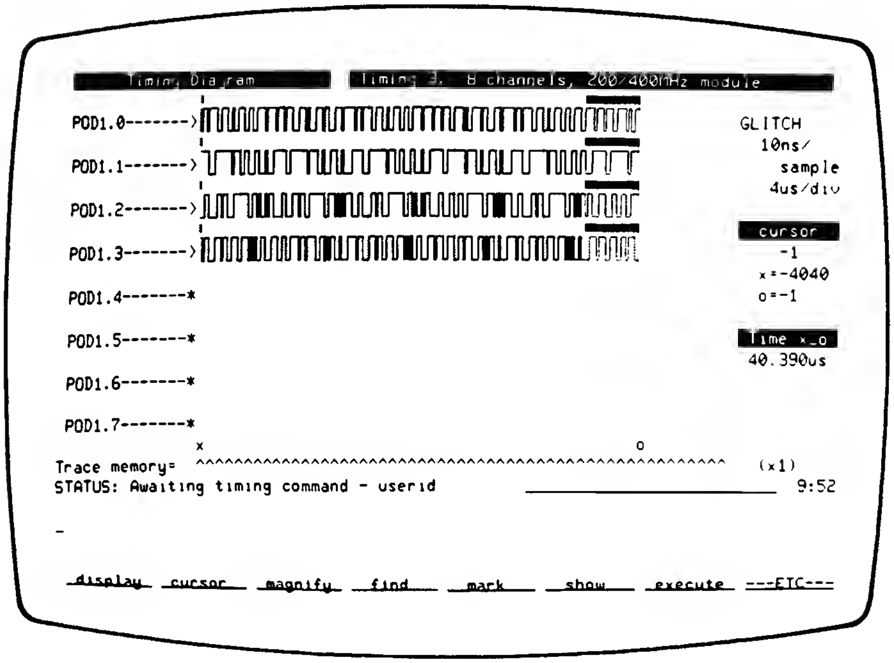
The timing diagram display specification softkeys consist of the utility and prompt softkeys, and the *display*, *magnify*, *cursor*, *indicate*, *find*, *mark*, *diagram*, and *<ROLL>* softkeys. The utility softkeys contained in the timing diagram display specification consist of the *show*, *sample*, *execute*, *halt*, *configure*, *copy*, *end*, and *---ETC---* softkeys. These softkeys are discussed in detail in chapter 4. The prompt softkeys are also discussed in chapter 4. The timing diagram display specification-specific softkeys are described in the following paragraphs. Although the *mode* utility softkey is not available in the timing diagram display specification, it is discussed briefly to introduce the format of the timing diagram display specification for each mode.

The *mode* softkey in the format specification and trace specification allows you to choose one of the four operating modes: wide sample mode, glitch capture mode, dual threshold mode, and fast sample mode. The default timing diagram display for each of these modes is shown in figures 7-1 through 7-4, respectively. The different modes are discussed in chapter 4 under "Utility Softkeys".



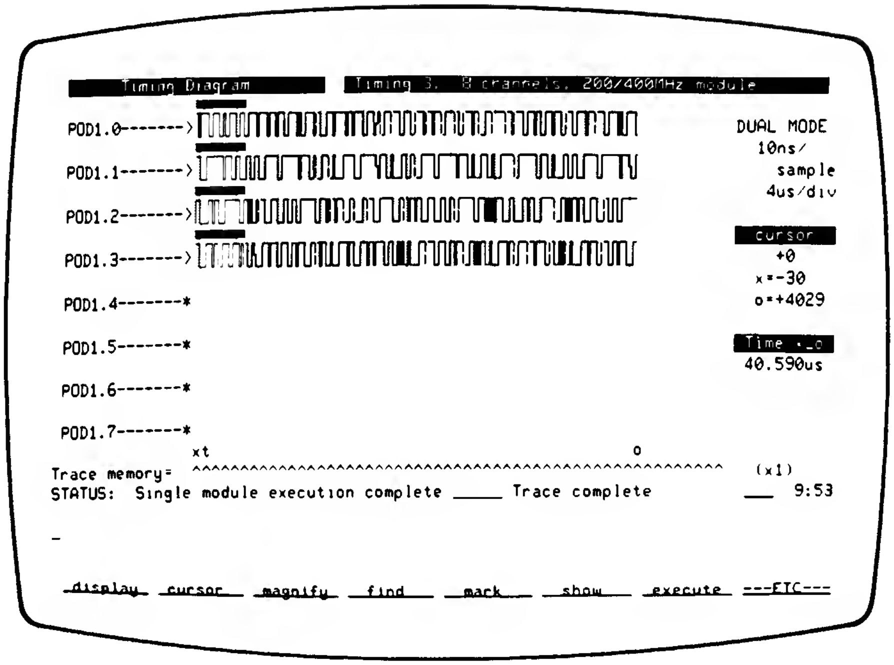
```
Timing Diagram                 Timing 3.  8 channels, 200/400MHz module

POD1.0-------->                                              WIDE MODE
                                                               5ns/
POD1.1-------->                                                 sample
                                                               2us/div
POD1.2-------->
                                                              cursor
POD1.3-------->                                                  +0
                                                               x=-30
POD1.4-------->                                                o=+4029

POD1.5-------->                                              Time x_o
                                                             20.2950us
POD1.6-------->

POD1.7-------->
               xt                                  o
Trace memory= ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^   (x1)
STATUS:  Single module execution complete ____ Trace complete      ___  9:50

-
```

Figure 7-1. Wide__Sample Default Timing Diagram Display



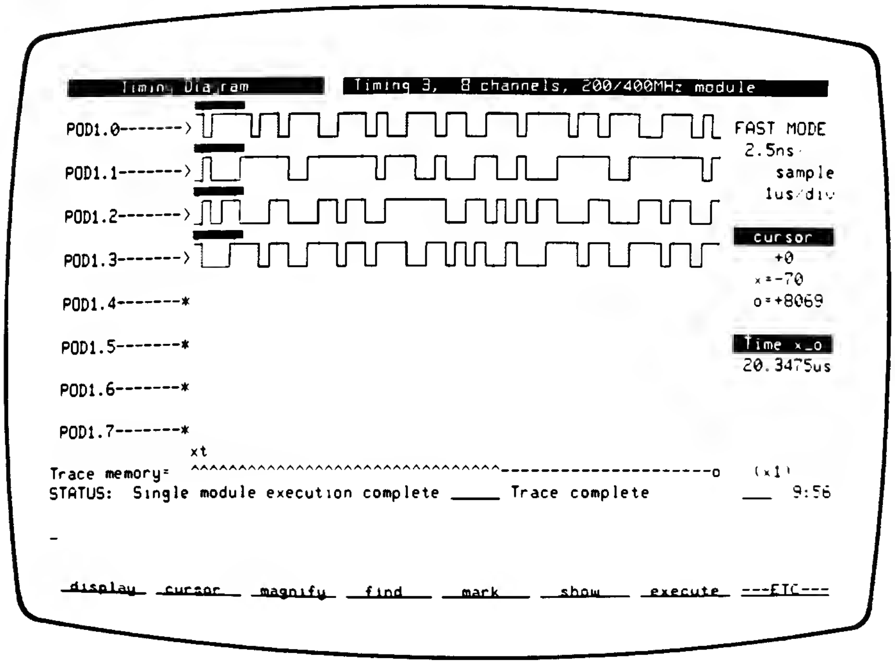
```
Timing Diagram                 Timing 3.  8 channels, 200/400MHz module

POD1.0-------->                                              GLITCH
                                                              10ns/
POD1.1-------->                                                 sample
                                                               4us/div
POD1.2-------->
                                                              cursor
POD1.3-------->                                                  -1
                                                               x=-4040
POD1.4-------*                                                 o=-1

POD1.5-------*                                               Time x_o
                                                             40.390us
POD1.6-------*

POD1.7-------*
               x                                   o
Trace memory= ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^   (x1)
STATUS: Awaiting timing command - userid       ____________________  9:52

-
```

Figure 7-2. Glitch__Capture Default Timing Diagram Display



```
 Timing Diagram          Timing 3,  8 channels, 200/400MHz module
POD1.0-------)                                                  DUAL MODE
                                                                 10ns/
POD1.1-------)                                                     sample
                                                                  4us/div
POD1.2-------)
                                                                 cursor
POD1.3-------)                                                     +0
                                                                 x=-30
POD1.4-------*                                                   o=+4029

POD1.5-------*                                                   Time x_o
                                                                 40.590us
POD1.6-------*

POD1.7-------*
              xt                                  o
Trace memory= ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^      (x1)
STATUS:  Single module execution complete _____ Trace complete          ___  9:53

-

display  cursor  magnify  find  mark  show  execute  ---ETC---
```

Figure 7-3. Dual__Threshold Default Timing Diagram Display



```
 Timing Diagram          Timing 3,  8 channels, 200/400MHz module
POD1.0-------)                                                  FAST MODE
                                                                 2.5ns
POD1.1-------)                                                     sample
                                                                  1us/div
POD1.2-------)
                                                                 cursor
POD1.3-------)                                                     +0
                                                                 x=-70
POD1.4-------*                                                   o=+8069

POD1.5-------*                                                   Time x_o
                                                                 20.3475us
POD1.6-------*

POD1.7-------*
              xt
Trace memory= ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^---------------------o      (x1)
STATUS:  Single module execution complete _____ Trace complete          ___  9:56

-

display  cursor  magnify  find  mark  show  execute  ---ETC---
```

Figure 7-4. Fast__Sample Default Timing Diagram Display

DISPLAY SOFTKEY

The *display* softkey allows you to set up any desired arrangement of traces on the timing diagram display. Up to 32 traces can be included in your display command although only 16 can be shown on screen at one time. When the *display* softkey is pressed, the softkey label line will show the following selections:

modify default blank <COMPARE> <DISPLAY> POD1 POD1upper POD1lower

NOTE

> If labels have been assigned in the format specification, the last softkey on the line will be ---*ETC*---. By pressing ---*ETC*---, additional labels will appear on a new line of softkey names. If more than one line is required to list all of the labels, press ---*ETC*--- again to call the next line of softkey names.

'display' 'modify'

The *modify* softkey allows you to call the present display specification to the command line for editing and reformatting of the timing diagram. The default display command is:

display POD1 then default_bits_in pod_1

Suppose you want to view the traces from POD1.0 through POD1.3 and you want maximum separation between these channels on screen. Move the cursor under the *t* in *then* using the <--, -->, or **TAB** keys. Use the **DELETE CHAR** key to delete *then default_bits_in pod_1* from the command line. Press the '.' key in response to the *<DOT>* prompt, then the **0** key in response to the *<BIT#>* prompt. Press the *then* and *blank* softkeys. Now press the *then* and *POD1* softkeys and answer the *<DOT>* and *<BIT#>* prompts again. Repeat the above procedure until all four bits are listed, with a blank between each. Press the **RETURN** key and observe that the timing diagram display appears as shown in figure 7-5.

'display' 'default'

This entry calls for the default display (identifying the source by probe pod and bit number) of each selected trace.

'display' 'blank'

This entry calls for displaying a blank trace at the selected location. Blank traces are used for separating groups of traces on the display.

'display' 'compare or <COMPARE>'

The *compare* entry calls for displaying activity from a bit in a compare file. The *<COMPARE>* prompt indicates that no compare file has been defined. You can define a compare file in the post_process specification. Then you can call for displaying activity from bits in the compare file, and you can arrange those displays beside displayed bits taken from your present system under test.

'display' '<DISPLAY>'

This prompt indicates that you can alternate the display between the label names assigned to each bit and the absolute source of each bit on screen by pressing the *display* and **RETURN** keys.

'display' 'POD1 (or any other label)'

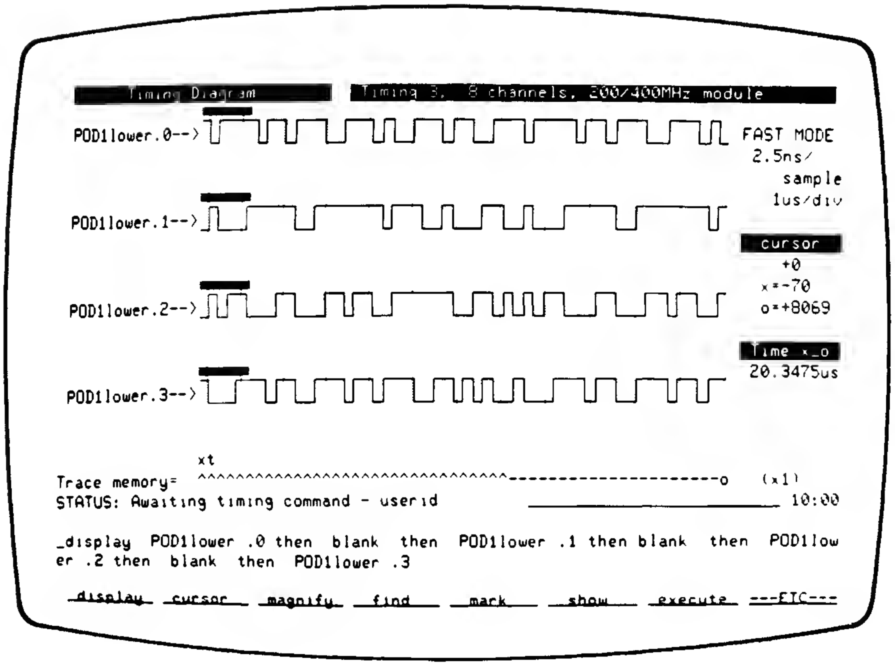
The remaining *display* softkeys have labels defined in the format specification. Use them to format the timing diagram for display of the traces desired.

MAGNIFY SOFTKEY

The *magnify* softkey allows the trace to be viewed normally (x1) or to be expanded by a factor of 10 (x10) or 100 (x100). It also provides for turning the magnify indicator *on* or *off*. The magnify indicator is the horizontal bar above each line that rides along with the cursor as the cursor is moved. The width of the bar indicates how much of the trace memory will be on screen in the next higher magnification (x10 or x100).

CURSOR SOFTKEY

Pressing the *cursor* softkey changes the name of the key to inverse video and allows you to move the cursor across the trace on the screen by using the --> and <-- keys. When the cursor moves, the number under the word "cursor" on the right side of the screen changes. This number denotes the position of the cursor with respect to the trigger (always on line 0). When the cursor reaches the end of the display, the --> and <-- keys cause the display to move past the cursor, unless the end of memory has been reached. Using the **SHIFT** key with the --> and <-- keys has the same effect. The up-carats (^) in the "Trace Memory=" line will move to indicate the changing position of the display window in memory. When the cursor softkey label is not shown in inverse video, the --> and <-- keys can move the cursor along the command line.



```
                xt
Trace memory=  ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^---------------------o    (x1)
STATUS: Awaiting timing command - userid                ____________________ 10:00

_display  POD1lower .0 then  blank  then  POD1lower .1 then blank  then  POD1low
er .2 then  blank  then  POD1lower .3
```

Figure 7-5. 8-Channel Timing Diagram Modify Display Example

INDICATE SOFTKEY

The *indicate* softkey can be used to obtain an indication of the signal levels present at the cursor. When the *indicate levels_at_cursor* option is *on*, an h, l, m, or g will appear at the end of each label line. These letters indicate whether the signal at the cursor is high, low, middle, or includes a glitch. By selecting *off*, the letter is replaced by an arrow.

The timing analyzer presents a field called 'Time x_o' on screen that shows information about the interval between the samples marked with *x* and *o*. You can select the kind of information to be shown in this field by using the *indicate* softkey. The default selection causes this field to indicate the time interval between *mark_x* and *mark_o*. You can also have this field indicate the number of *mark_abcd* samples between *mark_x* and *mark_o*.

By making a series of repetitive measurements of a particular time interval, the timing analyzer can provide additional information. If the interval is varying, the timing analyzer can gather statistics to show the minimum interval, maximum interval, mean interval, and the standard deviation (a measure of the dispersion) of the sampled interval. This information is useful when characterizing circuitry or system interactions. The results may show that the system is not working correctly, or they may indicate time interval variations larger than expected, requiring design changes.

The *indicate* softkey adds statistical information to the interval-measurement field, when desired. The *max_min* parameter adds two items to the interval-measurement field showing the maximum and minimum time periods or marks counted during a series of repetitive executions of a measurement.

The *mean_stdv* parameter is available for the *indicate* softkey. It calls for fields showing the average and the standard deviation from average of the time periods or marks counted during a series of repetitive measurements.

FIND SOFTKEY

The *find* softkey allows you to bring the trigger, any mark, or any timing pattern or duration which could have a mark assigned to it to the display and center it on screen, if possible. This capability brings these points to the screen quickly, without the need to hold down the --> or <-- keys to find them.

MARK SOFTKEY

The *mark* softkey allows you to identify or highlight samples on the timing diagram which meet specifications of your choice. Your *mark* assignments can be processed on the timing diagram for statistical computations and to help you locate events quickly. Once defined, you can turn *on* or *off* the display of any mark to aid in display interpretation, without upsetting its definition or the definitions of any of the other marks. The *named* parameter allows you to assign a descriptive name to each mark for display in a trace list. These names cannot be displayed in the timing diagram.

When you press the *mark* softkey, the following line of softkey selections appears on screen:

_______ ___*x*___ ___*o*___ ___*a*___ ___*b*___ ___*c*___ ___*d*___ _______

The *x* and *o* markers can identify only one sample in the trace memory. The *a*, *b*, *c*, and *d* markers can identify as many samples in memory as meet their specifications. The capabilities of each of these selections is described in the following paragraphs.

'mark_<xoabcd>' 'off' ['default']

The *off* entry alone turns off display of the selected mark without affecting its *mark* specification, and without affecting the display of any of the other markers. If you add the *default* parameter, your command will default the selected mark specification in the following way:

```
        mark_x will be assigned to sample -0030
        mark_o will be assigned to sample +4029
        mark_<abcd> will be undefined
```

'mark__<xoabcd>' 'on'

This entry turns on display of the mark you specify according to its present *mark* specification. The timing analyzer can process up to 511 marked locations for display.

'mark__<xo>' 'on__trigger' ['named' ....]

This entry allows you to place the selected *x* or *o* mark on the sample that was recognized as the trigger during the trace measurement.

'mark__<xo>' 'on__sample' <number> ['named' ....]

This entry places the selected *x* or *o* mark on any sample identified by its line number in memory.

'mark__<xo>' 'on__cursor' ['named' ....]

This entry assigns either the *x* or *o* mark to the position of the cursor. With this command, you can use the cursor to move the selected mark to any sample on the display with the keyboard <--, -->, and/or **ROLL** keys.

'mark__<xo>'

This entry marks the sample specified by the cursor with the selected mark.

'mark__<xo>' 'on__first__occurrence__of' 'entering/leaving/any__glch'

<LABEL> =/<> <VALUE> ['after' ....] ['named' ....]

Use this syntax to assign the *x* and/or *o* markers to events found in the trace memory. When you make this entry, the timing analyzer will search its memory and assign the specified mark to the sample where your specification is met. You can specify an occurrence to be found on a single probe bit or on a combination of probe bits. You can further specify that the occurrence must be found following some other reference point.

This command can be used to make a measurement of the time a selected line in your system remains true. By assigning *mark__x* to the sample where that line enters the true state and *mark__o* to the sample where that line leaves the true state for the first time after *mark__x*, the 'Time x_o' field on the timing diagram will show the duration of the true state.

'mark__<xo>' 'on__first__occurrence__of' 'greater__than/less__than'

<TIME>__of <LABEL> =/<> <VALUE> ['after' ....]

['named' ....]

This syntax assigns the *x* and/or *o* markers to samples in the trace memory where a value was unchanged for more than or less than a specified period of time. When you enter this command, the timing analyzer will search its trace memory and assign the specified mark to the first sample where your duration specification is met. You can specify a condition on a single bit or on a combination of bits. You can further specify a condition that must be found following a selected reference point.

This command can be used to find where a control line remains in the low state for an extended period of time. By assigning *mark__x* to the sample where this low state first exceeds the specified duration, and *mark__o* to the sample where the control line switches back to the high state, you can quickly find this condition and measure its duration.

'mark__<abcd>' 'on__all__occurrences__of'
'entering/leaving/greater__than/less__than/any__glch'

The *a*, *b*, *c*, and *d* markers allow you to mark up to four different conditions to be found as many times as they occur in the trace memory. As with the *x* and *o* markers, you can mark samples where bits or sets of bits entered or left values or where those bits retained their values for *greater__than* or *less__than* specified time periods. You can use these four *mark* resources to find all of the events of a particular kind so that they can be called to the display or counted in statistical calculations. You can further specify that any of these marks be assigned only after the occurrence of some event of reference in the trace memory. For example:

*mark a on__all__occurrences__of entering POD1 * 3AH*

after cursor named DISPLAY

DIAGRAM SOFTKEY

The *diagram* softkey allows you to select either the 8-channel or 16-channel format for viewing on screen. To make the selection, press the *diagram* softkey. The *8__chanls* and *16__chanls* softkeys appear on the softkey label line. Press the desired softkey, and then RETURN. The display will show the number of channels selected.

<ROLL> SOFTKEY

The *<ROLL>* softkey reminds you that the cursor position can be changed either by typing in the sample number, relative to the trigger, where you want the cursor located and pressing the RETURN key, or by pressing the SHIFT and --> or <-- keys, simultaneously.

Chapter 8

TRACE LIST DISPLAY SPECIFICATION

INTRODUCTION

This chapter describes the menu of softkeys that are specific to the trace list display specification. These softkeys, along with the utility softkeys described in chapter 4 are used to set up the trace list display specification for the timing analyzer. The softkeys that are included in this menu are described in detail on the following pages. Syntax diagrams for the more complex softkeys are shown in Appendix A.

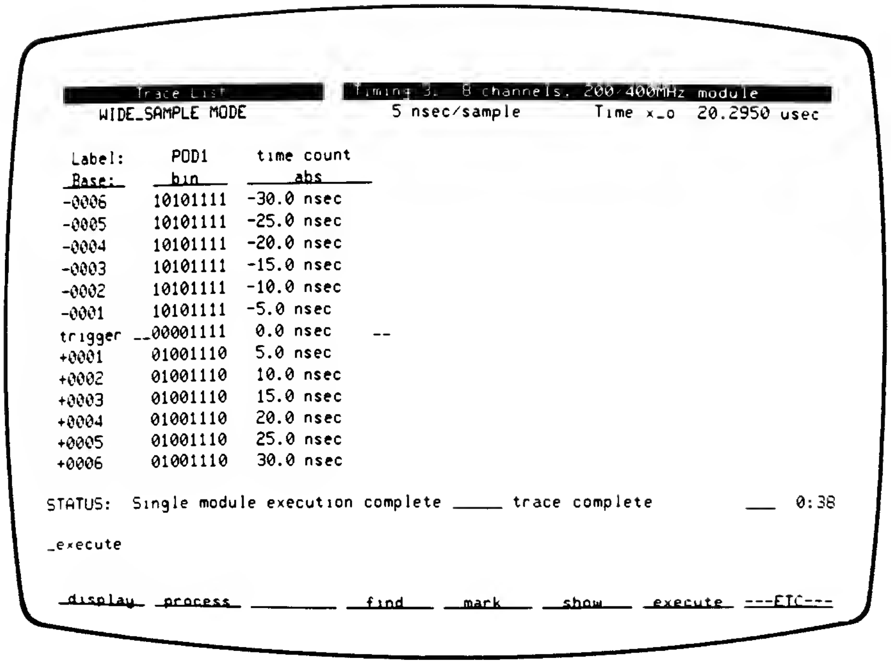
The trace list display specification and some of the options will vary, depending on whether the instrument is set up for 8- or 16-channel operation. Eight-channel operation is discussed in this chapter unless there are basic operational differences. In that case, both operational concepts will be discussed. The 8-channel default trace list display specification is shown in figure 8-1. A second line of softkey labels (shown below) is available by pressing the *---ETC---* softkey.

_____ *sample indicate <ROLL> configure copy end ---ETC---*

TRACE LIST DISPLAY SPECIFICATION SOFTKEYS

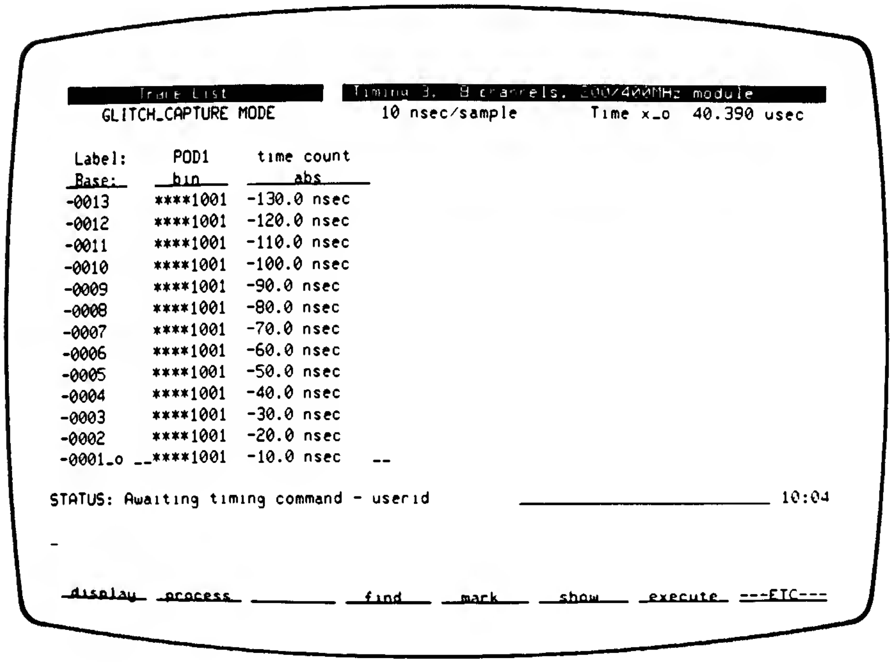
The trace list display specification softkeys consist of the utility and prompt softkeys, and the *display*, *find*, *mark*, *indicate*, *process*, and *<ROLL>* softkeys. The utility softkeys contained in the trace list display specification consist of the *show*, *sample*, *execute*, *halt*, *configure*, *copy*, *end*, and *---ETC---* softkeys. These softkeys are discussed in detail in chapter 4. The prompt softkeys are also discussed in chapter 4. The trace list display specification-specific softkeys are described in the following paragraphs. Although the *mode* utility softkey is not available in the trace list display specification, it is discussed briefly to introduce the format of the trace list display specification for each mode.

The *mode* softkey in the format specification and trace specification allows you to choose one of the four operating modes: wide sample mode, glitch capture mode, dual threshold mode, and fast sample mode. The default trace list display for each of these modes is shown in figures 8-1 through 8-4, respectively. The different modes are discussed in chapter 4 under "Utility Softkeys".



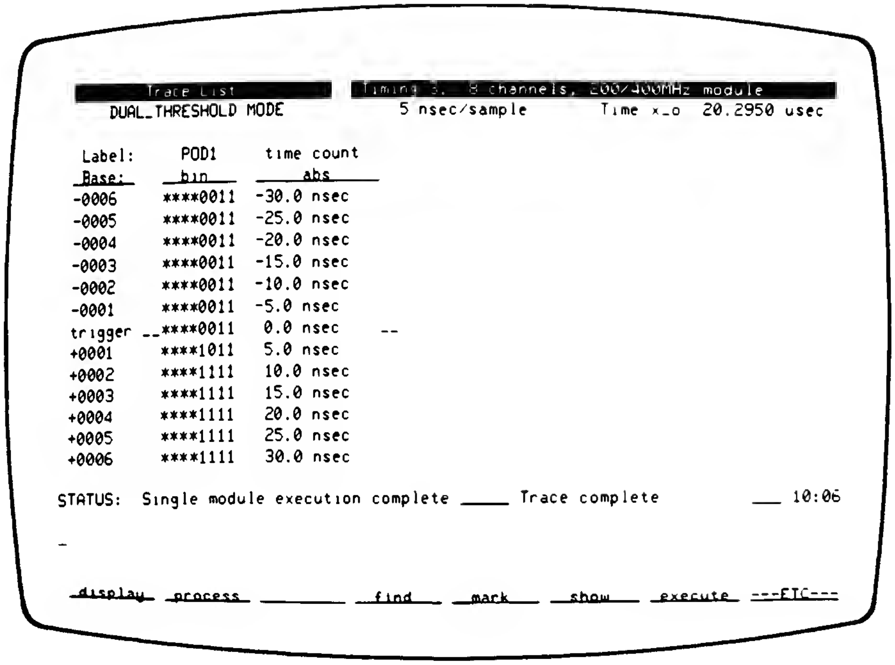
```
      Trace List            Timing 3.  8 channels, 200/400MHz module
   WIDE_SAMPLE MODE                5 nsec/sample        Time x_o  20.2950 usec

 Label:     POD1     time count
 Base:      bin         abs
-0006     10101111  -30.0 nsec
-0005     10101111  -25.0 nsec
-0004     10101111  -20.0 nsec
-0003     10101111  -15.0 nsec
-0002     10101111  -10.0 nsec
-0001     10101111  -5.0 nsec
trigger __00001111   0.0 nsec     --
+0001     01001110   5.0 nsec
+0002     01001110   10.0 nsec
+0003     01001110   15.0 nsec
+0004     01001110   20.0 nsec
+0005     01001110   25.0 nsec
+0006     01001110   30.0 nsec

STATUS:  Single module execution complete _____ trace complete          ___  0:38

_execute

 display   process  ________  find    mark     show    execute   ---ETC---
```

Figure 8-1. Wide__Sample Default Trace List Display



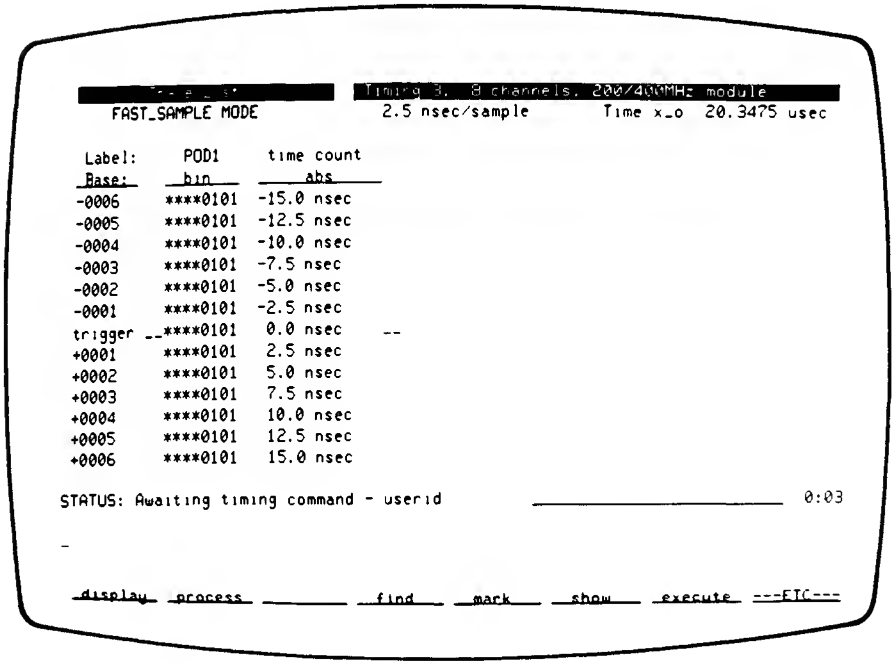
```
      Trace List            Timing 3.  8 channels, 200/400MHz module
   GLITCH_CAPTURE MODE            10 nsec/sample        Time x_o  40.390 usec

 Label:     POD1     time count
 Base:      bin         abs
-0013     ****1001  -130.0 nsec
-0012     ****1001  -120.0 nsec
-0011     ****1001  -110.0 nsec
-0010     ****1001  -100.0 nsec
-0009     ****1001  -90.0 nsec
-0008     ****1001  -80.0 nsec
-0007     ****1001  -70.0 nsec
-0006     ****1001  -60.0 nsec
-0005     ****1001  -50.0 nsec
-0004     ****1001  -40.0 nsec
-0003     ****1001  -30.0 nsec
-0002     ****1001  -20.0 nsec
-0001_o __****1001  -10.0 nsec    --

STATUS: Awaiting timing command - userid        ________________________ 10:04

_

 display   process  ________  find    mark     show    execute   ---ETC---
```

Figure 8-2. Glitch__Capture Default Trace List Display



```
 Trace List                  Timing 3.  8 channels, 200/400MHz module
  DUAL_THRESHOLD MODE            5 nsec/sample       Time x_o  20.2950 usec

 Label:     POD1     time count
 Base:      bin         abs
-0006     ****0011  -30.0 nsec
-0005     ****0011  -25.0 nsec
-0004     ****0011  -20.0 nsec
-0003     ****0011  -15.0 nsec
-0002     ****0011  -10.0 nsec
-0001     ****0011  -5.0 nsec
trigger --****0011   0.0 nsec    --
+0001     ****1011   5.0 nsec
+0002     ****1111   10.0 nsec
+0003     ****1111   15.0 nsec
+0004     ****1111   20.0 nsec
+0005     ****1111   25.0 nsec
+0006     ****1111   30.0 nsec

STATUS:  Single module execution complete ____ Trace complete          ___ 10:06

-

 display  process  ________  find   mark   show   execute  ---ETC---
```

Figure 8-3. Dual__Threshold Default Trace List Display



```
 Trace List                  Timing 3.  8 channels, 200/400MHz module
  FAST_SAMPLE MODE             2.5 nsec/sample       Time x_o  20.3475 usec

 Label:     POD1     time count
 Base:      bin         abs
-0006     ****0101  -15.0 nsec
-0005     ****0101  -12.5 nsec
-0004     ****0101  -10.0 nsec
-0003     ****0101  -7.5 nsec
-0002     ****0101  -5.0 nsec
-0001     ****0101  -2.5 nsec
trigger --****0101   0.0 nsec    --
+0001     ****0101   2.5 nsec
+0002     ****0101   5.0 nsec
+0003     ****0101   7.5 nsec
+0004     ****0101   10.0 nsec
+0005     ****0101   12.5 nsec
+0006     ****0101   15.0 nsec

STATUS: Awaiting timing command - userid          ____________________  0:03

-

 display  process  ________  find   mark   show   execute  ---ETC---
```

Figure 8-4. Fast__Sample Default Trace List Display

DISPLAY SOFTKEY

The *display* softkey allows you to set up any desired arrangement of information in the trace list display. When the *display* softkey is pressed, the softkey label line will show the following selections:

modify default time_cnt <COMPARE> mark_name POD1 POD1upper POD1lower

'display' 'modify'

The *modify* softkey allows you to call the present display specification to the command line for editing and reformatting of the display. The default display command is:

display POD1 then time_count absolute

Suppose you want to view the information from POD1 in hexadecimal values instead of the default binary values. Move the cursor under the *t* in *then* using the <--, -->, or TAB keys. Now press the **INSERT CHAR** key, and the *in_hex* softkey. Press the **INSERT CHAR** key again, and press **RETURN**. The trace list display will appear as shown in figure 8-5 instead of the default display shown in figure 8-1.

'display' 'default'

This entry calls for the default display (binary) of selected probe bits.

'display' 'time_cnt'

This entry calls for a column listing the time relationships of each of the samples listed. Absolute time count shows the time between each sample and the trigger sample. Relative time count shows the time from each sample to the next sample in memory.

'display' 'compare or <COMPARE>'

The *compare* entry calls for a column listing activity from a label in a compare file. The *<COMPARE>* prompt indicates that no compare file has been defined. You can define a compare file in the post_process specification. Then you can call for listings of activity from the compare file and compare them with activity derived from your present system under test.

'display' 'mark_name'

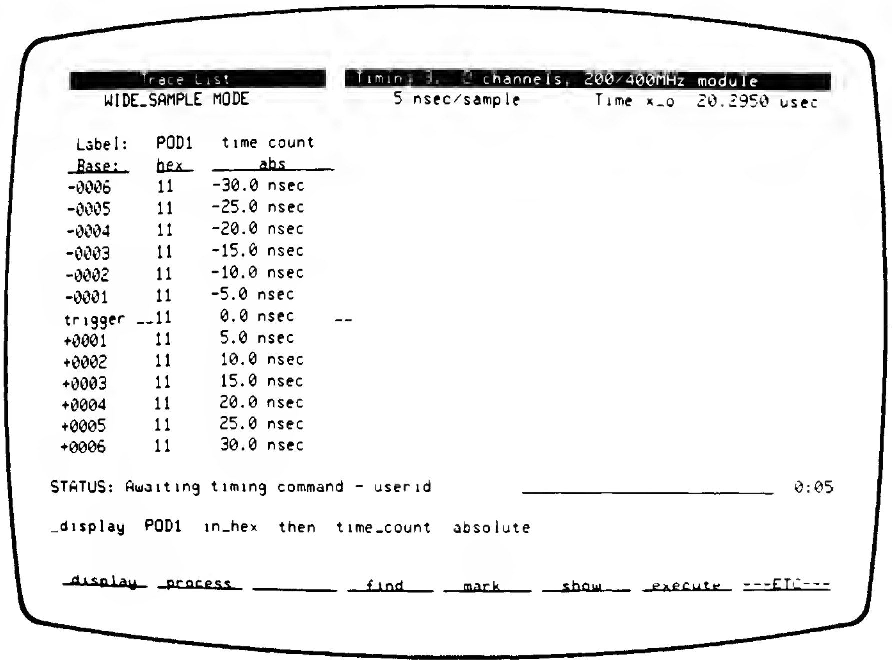
This entry calls for a column listing the names you assigned to identify marks. Each sample on the display that has a mark assigned to it will show the name of that mark in this column. Otherwise, this column will be blank.

'display' 'POD1 (or any other label)'

Use the remaining *display* softkeys to format the display for your convenience.

FIND SOFTKEY

The *find* softkey allows you to bring the trigger, any mark, or any timing pattern or duration which could have a mark assigned to it to the display, centered on screen if possible, without using the ROLL, PAGE, or 'up/down' arrow keys. Just press the *find* softkey and the softkey defining the point you want to view, and then press the RETURN key. The selected point will appear on screen.



```
        Trace List            Timing 3. 8 channels, 200/400MHz module
   WIDE_SAMPLE MODE                  5 nsec/sample       Time x_o  20.2950 usec

 Label:    POD1   time count
 Base:     hex        abs
-0006      11    -30.0 nsec
-0005      11    -25.0 nsec
-0004      11    -20.0 nsec
-0003      11    -15.0 nsec
-0002      11    -10.0 nsec
-0001      11    -5.0 nsec
trigger --11      0.0 nsec    --
+0001      11     5.0 nsec
+0002      11     10.0 nsec
+0003      11     15.0 nsec
+0004      11     20.0 nsec
+0005      11     25.0 nsec
+0006      11     30.0 nsec

STATUS: Awaiting timing command - userid                          0:05

_display  POD1  in_hex  then  time_count  absolute

 display  process            find    mark    show    execute  ---ETC---
```

Figure 8-5. 8__Channel Trace List Modify Display Example

MARK SOFTKEY

The *mark* softkey allows you to assign a mark to identify or highlight samples in the trace list that meet specifications of your choice. Your *mark* assignments can be processed for statistical computations and to help you locate events quickly. Once defined, you can turn on or off the display of any *mark* to aid in display interpretation, without upsetting its definition or the definition of any other *mark*. Each of your marks can have a name of your choice assigned to it for further identification. You can include a column listing these names in your trace list, if desired.

When you press the *mark* softkey, the following line of softkey selections appears on screen:

```
_______ ___x___ ___o___ ___a___ ___b___ ___c___ ___d___ _______
```

The *x* and *o* marks identify one sample each in the trace memory. The *a*, *b*, *c*, and *d* marks can identify as many samples as meet their specifications. The capabilities of each of these selections is described in the following paragraphs.

'mark__<xoabcd>' 'off' ['default']

The *off* entry alone turns off display of the selected mark without affecting its *mark* specification, and without affecting the display of any of the other marks. If you add the *default* parameter to your command, the analyzer will default the selected mark specification in the following way:

```
        mark_x will be assigned to sample -0030
        mark_o will be assigned to sample +4029
        mark_<abcd> will be undefined
```

'mark__<xoabcd>' 'on'

This entry turns on display of the mark you specify according to its present *mark* specification. The timing analyzer can process up to 511 marked samples for display.

'mark__<xo>' 'on__trigger' ['named' ....]

This entry places the selected *x* or *o* mark on the sample that was recognized as the trigger during the trace measurement.

'mark__<xo>' 'on__sample' <number> ['named' ....]

This entry places the selected *x* or *o* mark on any sample identified by its line number in memory.

'mark__<xo>' 'on__cursor' ['named' ....]

This entry assigns either the *x* or *o* mark to move along with the cursor. With this command, you can use the cursor to move the selected mark to any sample on the display with the keyboard **<--**, **-->**, and/or **ROLL** keys.

'mark__<xo>'

This entry marks the sample specified by the cursor with the specified mark.

'mark__<xo>' 'on__first__occurrence__of' 'entering/leaving/any__glch'

<LABEL> =/<> <VALUE> ['after' ....] ['named' ....]

This syntax assigns the *x* and/or *o* marks to events found in the trace memory. When you make this entry, the timing analyzer will search its memory and assign the specified mark to the sample where your specification is met. You can specify a change of value to be found on a single probe bit or on a combination of probe bits. You can further specify that the change of value must be found following some other reference point.

This command can be used to make a measurement of the time that a selected line in your system remains true. By assigning *mark__x* to the sample where that line enters the true state and *mark__o* to the sample where that line leaves the true state for the first time after *mark__x*, the 'Time x_o' field on the trace list will show the duration of the true state.

'mark__<xo>' 'on__first__occurrence__of' 'greater__than/less__than'

<TIME>__of <LABEL> =/<> <VALUE> ['after' ....]

['named' ....]

This syntax assigns the *x* and/or *o* marks to samples in the trace memory where a value remained the same for *greater__than* or *less__than* a specified period of time. When you enter this command, the timing analyzer will search its trace memory and assign the specified mark to the sample where your duration specification is first met. You can specify a condition to be found on a single bit or on a combination of bits. You can further specify a condition that must be found following a selected reference point.

This command can be used to find where a control line remains in the low state for an extended period of time. By assigning *mark__x* to the sample where this low state first exceeds the specified duration, and *mark__o* to the sample where the control line switches back to the high state, you can quickly find this condition and measure its duration.

'mark_<abcd>' 'on_all_occurrences_of'
'entering/leaving/greater_than/less_than/any_glch'

The *a*, *b*, *c*, and *d* marks allow you to identify up to four different conditions to be found and marked as many times as they occur in trace memory. As with the *x* and *o* marks, you can mark samples where bits or sets of bits entered or left values or where those bits retained their values for *greater_than* or *less_than* specified time periods. You can use these four *mark* resources to identify all of the events of a particular kind so that they can be called to the display or counted in statistical calculations. You can further specify that any of these marks be assigned only after the occurrence of some event of reference in the trace memory. For example:

*mark a on_all_occurrences_of entering POD1 **

3AH after cursor named DISPLAY

INDICATE SOFTKEY

The timing analyzer presents a field called 'Time x_o' on screen that shows information about the interval between the samples marked with *x* and *o*. You can select the kind of information to be shown in this field by using the *indicate* softkey. The default selection causes this field to indicate the time interval between *mark_x* and *mark_o*. You can also have this field indicate the number of *mark_abcd* samples between *mark_x* and *mark_o*.

By making a series of repetitive measurements of a particular time interval, the timing analyzer can provide additional information. If the interval is varying, the timing analyzer can gather statistics to show the minimum interval, maximum interval, mean interval, and the standard deviation (a measure of the dispersion) of the sampled interval. This information is useful when characterizing circuitry or system interactions. The results may show that the system is not working correctly, or they may indicate time interval variations larger than expected, indicating a need for design changes.

The *indicate* softkey adds statistical information to the interval-measurement field, when desired. The *max_min* parameter adds two items to the interval-measurement field showing the longest interval measured or most marks counted, and the shortest interval measured or least marks counted during a series of repetitive executions of a measurement.

The *mean_stdv* parameter is available for the *indicate* softkey. It calls for fields showing the average time or marks measured and the standard deviation from the average during a series of repetitive measurements.

PROCESS SOFTKEY

The *process* softkey allows you to set up display qualifications to obtain tracelists limited to data of interest. For example, you can use the timing analyzer to capture state activity in a very fast system. Then you can use the *process* command to provide a trace list of the states captured. The trace list will show each of the labels you selected, and the states that were found on those labels. The states can be shown in any one of the four standard number bases you select. The *process* command softkey offers several selections to get this display. The way to use these softkey selections are discussed in the following paragraphs. When the *process* softkey is pressed, the softkey label line will show the following selections:

modify marked greater sampled _____ *off* _____ _____

'process__for__data' 'modify'

This entry calls the present *process* specification to the command line of the display so that you can edit its content for a new *process* specification.

'process__for__data' 'marked'

This entry provides a trace list of only those transactions that meet your *mark* specifications. You can use this entry to see the points in memory where events of interest occurred.

You can obtain trace lists of state executions using this specification. Use the "mark_abcd" specifications to identify conditions where the data on selected lines or sets of lines remained stable for enough time to be recognized as stable states by the system under test. Then enter the specification to "Process for data marked". By using "mark" assignments, you can get a listing of only the samples with data that was stable long enough to meet your duration specifications. The limitation of using mark assignments is that the analyzer can only mark 511 samples so if you have more than 511 states in memory, they will not all be shown on the processed display.

'process__for__data' 'greater__than'

Use this entry to obtain a list of samples where patterns of your choice are unchanged for more than a specified period of time. For example, the entry *process__for__data greater__than 2.5 usec on POD1 .7* might give you a trace list with only one line on screen. This line would be the only place in memory where your specification was met (where the state of pod 1, bit 7 had not changed for 2.5 usec). This syntax can help you find places where conditions remained the same for inordinate periods of time.

You can use this softkey selection to obtain a trace list of state executions in a fast state machine. By setting up a *process* command that lists states lasting longer than the minimum time required to be recognized as states in the system under test, you can obtain a trace list of every state in the memory. This specification is not limited to 511 states like the "mark" assignments.

When using this method to obtain a trace list of state flow, you must select a sample rate that allows the timing analyzer to sample each state at least twice to avoid listing the transition states. This requirement limits the rate of state flow that can be captured by the timing analyzer. The fastest state flow rate the timing analyzer can capture is twice the timing analyzer sample period plus 3-ns skew. At the fastest sample periods, the timing analyzer can capture state flow from systems operating at the following rates:

1. With 5-ns sample period:

   2 * 5 ns + 3 ns = 13 ns

   This limits state capture to systems operating at 77 MHz or less.

2. With 2.5-ns sample period (fast sample mode):

   2 * 2.5 ns + 3 ns = 8 ns

   This limits state capture to systems operating at 125 MHz or less.

'process__for__data' 'sampled' NUMBER 'samples__before/after'

This *process* command allows you to display-qualify samples that were obtained at fixed times either before or after a reference event, and to identify that reference event as either a positive transition, a negative transition, or any transition on a bit you specify.

For example, you might set up a sample rate of 5 nsec, and then enter *process__for__data sampled 4 samples__after pos__transition__on POD1 .1* . The analyzer will present a list of only those samples that were obtained 20 nsec after each positive transition on bit 1 of pod 1.

You can use this specification to obtain a list of states executed in a fast system. Connect one of the probe pod channels to the data clock in the system under test. If the data is read on the positive-going edge, you can enter a process command such as "Process for data 1 cycle before positive transition on CLOCK channel". The timing analyzer will show a list of states that were captured one sample period before the timing analyzer found that the system "read-transition" had occurred. You can process for data "2 cycles before", or "5 samples before", or "5 samples after", or any other specification to see the setup and hold times of the states as well as the actual states that were executed in the system under test.

When using this method to obtain a list of state executions, the state must be present for a time of at least one sample period plus the 3-ns skew before the clock transition. This means if you are using a 5-ns sample period, then data must be present for at least 8 ns before the clock transition in the system under test.

The clock in the system under test must operate at a rate that guarantees it will be sampled true and sampled false at least one time each for each new state. When using the 5-ns sample period, each clock state must last at least 5-ns plus the 1.5-ns single-channel skew. Therefore the minimum clock period is 5 + 1.5 + 5 + 1.5 = 13 ns. The 13-ns clock period indicates that the maximum clock frequency of the system under test is 77 MHz.

When using the fast sample mode of measurement, the minimum setup time for each state is 5.5 ns before the clock transition, and the maximum clock frequency in the system under test is 125 MHz.

'process__for__data' 'off'

This is the default syntax for the *process* command. It display-qualifies every sample in the trace memory. Enter this command after you have finished using a special *process* selection and want to return to normal display processing.

<ROLL> SOFTKEY

The *<ROLL>* softkey prompt reminds you that the cursor position, can be changed (rolled) either by typing in the sample number where you want the cursor located and pressing the **RETURN** key, or by pressing the **ROLL UP**, **ROLL DOWN**, **NEXT PAGE**, **PREV PAGE**, or up-arrow/down-arrow keys to roll the screen up or down.

Chapter 9

POST_PROCESS SPECIFICATION

INTRODUCTION

This chapter describes the menu of softkeys that are specific to the post_process specification. These softkeys, along with the utility softkeys described in chapter 4 are used to set up the post_process specification for the timing analyzer. The softkeys that are included in this menu are described in the following pages. Syntax diagrams for the more complex softkeys are shown in Appendix A.

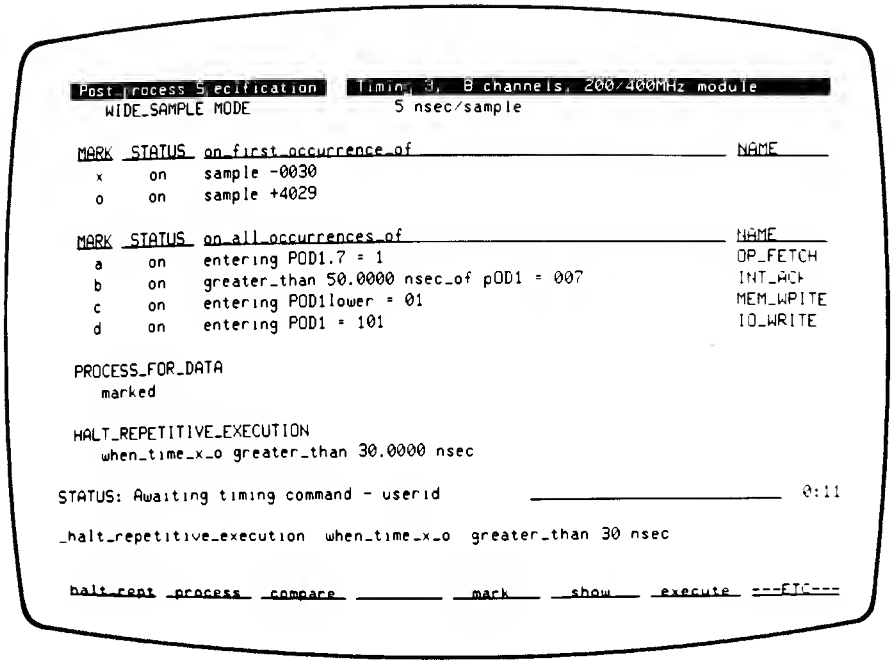
The post_process specification will vary, depending on whether the timing analyzer is set up for 8-channel or 16-channel operation. Eight-channel operation is discussed in this chapter, except where there are operational differences. In those cases, both operating configurations are discussed. The eight-channel default post_process specification is shown in figure 9-1.

A second line of softkey labels (shown below) is available by pressing the ---*ETC*--- softkey.

_______ _________ *default* _______ *configure* *copy* *end* ---*ETC*---

POST_PROCESS SPECIFICATION SOFTKEYS

The post_process specification softkeys consist of the utility and prompt softkeys, and the *halt_rept*, *process*, *compare*, and *mark* softkeys. The utility softkeys contained in the post_process specification consist of the *show*, *execute*, *default*, *configure*, *copy*, and *end* softkeys. These softkeys are discussed in chapter 4. The prompt softkeys are also discussed in chapter 4. The post_process specification-specific softkeys are described in the following paragraphs.



```
Post_ rocess S ecification     Timin  3,  8 channels, 200/400MHz module
   WIDE_SAMPLE MODE                 5 nsec/sample

MARK  STATUS  on_first_occurrence_of                              NAME
  x     on    sample -0030
  o     on    sample +4029

MARK  STATUS  on_all_occurrences_of                               NAME
  a     on    entering POD1.7 = 1                                 OP_FETCH
  b     on    greater_than 50.0000 nsec_of pOD1 = 007             INT_ACK
  c     on    entering POD1lower = 01                             MEM_WPITE
  d     on    entering POD1 = 101                                 IO_WRITE

PROCESS_FOR_DATA
   marked

HALT_REPETITIVE_EXECUTION
   when_time_x_o greater_than 30.0000 nsec

STATUS: Awaiting timing command - userid                          0:11

_halt_repetitive_execution  when_time_x_o  greater_than 30 nsec
```

Figure 9-1. Post__Process Specification Example

HALT__REPT SOFTKEY

The *halt__rept* softkey allows you to set up a variety of specifications to extend the triggering capability of the Timing Analyzer. The trigger function of the timing analyzer is a 1, 2, 5 function that is separate from the sample-rate function. It allows trigger recognition on a condition that is a multiple of 1, 2, or 5, such as 'longer than 500 usec', but it cannot trigger on other multiples, such as 'longer than 576 usec'. The post_processing capability of the timing analyzer does allow recognition of these kinds of conditions. You can select the repetitive execution measurement mode and enter the desired occurrence, such as 'longer than 576 ms', on which to halt the repetitive executions using the *halt__rept* softkey. When the *halt__rept* softkey is pressed, the softkey label line will show the following selections:

modify time marks sequence runs ______ off ______

Each of these selections offers a different capability for extending the timing analyzer triggering. These are all discussed in the following paragraphs.

`halt__repetitive__execution' `when__time__x__o'

Suppose you are testing a system which handshakes information between components. The handshake lines usually require a normal range of time to perform their functions, but occasionally the handshake of information requires an unusually long time. You would like to have the Timing Analyzer take a series of measurements of the data-transfer handshake lines and check the length of time required to handshake information. Each time the handshake is completed within the normal time period, you would like the Timing Analyzer to take a new trace. When the Timing Analyzer captures a trace in which the handshake takes an unusually long time to complete, you would like the Timing Analyzer to stop tracing and hold the captured data for your inspection. This softkey entry allows you to perform that kind of measurement.

Use mark *x* and *o* to define the start and end of the handshake sequence. The *halt__rept* and *time* softkeys will allow you to specify the trigger time period as a time *greater__than* or *less__than* the normal time period you enter. The timing analyzer will continue tracing until your specification is met (the time period you specify is exceeded).

`halt__repetitive__execution' `when__marks__x__o'

Suppose you are testing a system that normally interrogates a series of I/O lines several times during a certain period of program execution, but occasionally it seems that the analyzer does not interrogate those I/O lines often enough. You would like the analyzer to run a continuous series of traces, and count the number of times that the system under test interrogates the I/O lines during a period of program flow. Each time it finds a normal number of I/O interrogations, you would like the analyzer to take a new trace. When it finds too few I/O interrogations during the period of measurement time, you would like the timing analyzer to stop tracing and hold its data in memory for your investigation. This softkey selection will allow you to set up the timing analyzer for that kind of trace.

Use mark *x* and *o* to define sample numbers in the trace memory (such as sample 0 and 4000) and select a sample rate that corresponds with your *x* and *o* samples to give you the desired time period to look for I/O interrogations. Mark the I/O interrogations using the *mark_<abcd>* resources. This syntax will allow you to capture and hold a trace of information where the number of *mark_<abcd>* occurrences is either *less_than* or *greater_than* a number of your choice.

'halt_repetitive_execution' 'when_sequence_x_o'

Suppose you are testing a system that normally performs a complex routine by activating a series of asynchronous lines in a particular sequence. Suppose the routine occasionally fails, and you would like to see a trace of the sequence of activating those asynchronous lines when it occurs in incorrect order. With this syntax, you can set up the analyzer to take a trace and check the sequence of events on the asynchronous lines. Each time the sequence occurs in normal order, the analyzer will begin a new measurement. If the analyzer ever finds a sequence that was performed in an abnormal order, it will stop tracing and hold the data with the abnormal sequence in memory for your inspection.

Use all of the *mark_<xoabcd>* resources, as required, to identify activities that make up events to be found in the sequence of interest. The analyzer can check for occurrences or for "not-occurrences" at any points in your specified sequence.

'halt_repetitive_execution' 'when_runs_equals'

Suppose you are performing a statistical trace to derive the average number of occurrences of a certain type during a fixed period of program execution. You would like the average to be derived from exactly 100 traces. This syntax will allow you to set up the timing analyzer to make exactly 100 traces and then stop tracing. You can define marks to identify the time period to be measured and the occurrences of interest. When you execute your measurement, the analyzer will perform the specified number of traces, compute your statistical information, and automatically stop tracing after it completes the specified number of traces, and it will hold the statistical information for your inspection.

'halt_repetitive_execution' 'off'

This is the default syntax for an *execute repeat* command. If you have entered one of the previous selections for extending trigger capability, this command will return your analyzer to normal processing. Enter this command to cancel the special trigger condition specified.

PROCESS SOFTKEY

The *process* softkey allows you to set up special display qualifications for the data in memory so that you can obtain displays limited to data of interest in a trace list. When the *process* softkey is pressed, the softkey label line will show the following selections:

modify marked greater sampled ______ *off* ______ ______

The capabilities offered by each of these selections are described in the following paragraphs.

`process__for__data' `modify'

This entry calls the present *process* command to the command line of the display so that you can edit its content for a new *process* command.

`process__for__data' `marked'

Suppose you would like to see a trace list of only those lines in memory which begin a particular kind of transaction. You can define a mark or marks to identify those transactions, and then use this entry to call only the marked lines to the screen.

`process__for__data' `greater__than'

Suppose you would like to see only those samples where a certain condition remained in effect for a particularly long period of time. For example, you might enter *process__for__data greater__than 2.5 usec on POD1 .7* . When you press the RETURN key, your display might show just one line number, such as +3477, along with its data. This is the only line in the trace memory where bit 7 of pod 1 was unchanged for 2.5 usec. This syntax will help you locate places in memory where selected conditions might remain unchanged for inordinate periods of time.

`process__for__data' `sampled' <number> `samples__before/after'

Suppose you would like to obtain a trace list that shows the conditions of a set of lines 20 nsec after the occurrence of each clock pulse during a measurement. You might set up a sample rate of 5 nsec, and then enter the following *process* command:
process__for__data sampled 4 samples__after pos__transition__on POD1 .1 .
(This assumes that the clock is connected to pod 1, and you want to display-qualify the state that occurs 20 nsec after the positive transition on the clock line.

The syntax of this *process* command allows you to display-qualify samples that occurred either before or after the event of reference, and to identify that event as either a positive transition, a negative transition, or any transition (either positive or negative).

`process_for_data' `off'

This is the default syntax for the *process* command. It display-qualifies every sample stored in trace memory so that all lines can be displayed on screen. Enter this command after you have finished using a special *process* selection and want to return to normal display processing.

COMPARE SOFTKEY

The *compare* softkey allows you to specify a configuration that you previously stored (with data) to be used by the timing analyzer when making comparison measurements. Enter *compare_file_is* and the name of the stored configuration. When you press the **RETURN** key, the timing analyzer will show the name of the file you entered as the present compare file. Then you can obtain displays of data from that file on timing diagrams and trace lists beside data from the present measurement configuration.

Certain attributes in your *compare_file* must match attributes in your present measurement setup. The *sample rate*, *trigger position*, and *mode* attributes, for example, must match in order to obtain displays comparing data from both sources. If they do not match, the timing analyzer will show an asterisk beside the *compare_file* labels on a timing diagram, or asterisks in place of values in the *compare_file* columns on a trace list. To obtain comparison displays, change the appropriate attributes in your measurement specification and execute a new trace. You can easily do this by using the *configure* and *load_from* softkeys to load the configuration of the *compare_file*. Then execute a new trace.

MARK SOFTKEY

The *mark* softkey allows you to locate and highlight events of interest in the timing trace memory by identifying the specifications met by those events. Your *mark* specifications can be used by the timing analyzer for computations, to process display-qualification, and to help you quickly locate events either on a timing diagram or in a list of trace activity. The parameter *named* allows you to assign descriptive names to the marks for display in a column beside the associated mark on a trace list. Once defined, you can turn off the display of any mark to aid in display interpretation without upsetting the definitions assigned to any of the marks.

When you press the *mark* softkey, the following line of softkey selections appears on screen:

```
_______ ___x___ ___o___ ___a___ ___b___ ___c___ ___d___ _______
```

The *x* and *o* markers will identify one sample each in the trace memory. The *a*, *b*, *c*, and *d* markers will identify all of the samples that meet their specifications in the trace memory. The capabilities of each of these selections is described in the following paragraphs.

`mark__<xoabcd>' `off' [`default']

The *off* entry alone turns off display of the mark you specify without disturbing its *mark* specification, or display of any of the other marks. If you add the *default* parameter to your command, it will affect the specifications in the following ways:

mark_x will be assigned to sample -0030.
mark_o will be assigned to sample +4029.
mark_<abcd> will be undefined.

`mark__<xoabcd>' `on'

This entry turns on display of the mark you specify according to its present *mark* specification. The timing analyzer can process up to 511 marks on the display.

`mark__<xo>' `on__trigger' [`named' ....]

This entry allows you to place the desired *x* or *o* mark on the sample that was recognized as the trigger during the trace measurement.

`mark__<xo>' `on__sample' <number> [`named' ....]

This entry places the selected *x* or *o* mark on any sample identified by its line number in memory.

`mark__<xo>' `on__cursor' [`named' ....]

Use this entry to assign the *x* or *o* mark to the position of the cursor on the display. You can use the cursor to move the mark to any sample of your choice by using the keyboard <--, -->, and/or ROLL keys.

`mark__<xo>'

Use this entry to release the specified mark from the cursor, leaving it on the sample presently marked.

`mark__<xo>' `on__first__occurrence__of' `entering/leaving/any__glch'

<LABEL> =/<> <VALUE> [`after' ....] [`named' ....]

Suppose you are going to make a measurement of the time that a status line in your system remains in the high state. You can use this command to assign *mark__x* to the first sample where the line enters the high state and *mark__o* to the first sample where the line leaves the high state after *mark__x*.

By using this syntax, the timing analyzer will locate the samples that meet your specifications and show the interval on screen. You can assign a mark to an occurrence on a single probe bit or a combination of bits. You can also specify that the mark be assigned on the first event to meet your specifications after one of several reference events.

`mark__<xo>' `on__first__occurrence__of' `greater__than/less__than'

<TIME>__of <LABEL> =/<> <VALUE> [`after' ....]

[`named' ....]

Suppose you are going to make a measurement of system activity after a certain control line has remained in the low state for a particular period of time. You can use this entry to assign *mark__x* to the sample where the time specification is met, and then assign *mark__o* to identify the sample where this control line switches back to the high state, ending your measurement.

You can specify that either mark be assigned to the duration of a condition on any bit. You can further specify that the mark be assigned on the first occurrence after one of several reference events in the trace.

`mark__<abcd>' `on__all__occurrences__of'
`entering/leaving/greater__than/less__than/any__glch`

This syntax allows you to assign up to four marks *<abcd>* to identify four different conditions to be found in trace memory. These assignments differ from the *mark__<xo>* assignments in that they will identify all occurrences that meet their specifications, not just the first occurrences. You can use these four resources to identify events of interest to be counted in statistical calculations.

Like the *mark__<xo>* resources, you can assign these marks to identify samples where values enter or leave labeled bits or sets of bits as well as durations of values on those bits. You can further specify that any of these marks only be assigned after some event of reference in the trace. For example:

mark__a on__all__occurrences__of entering POD1 =

3AH after cursor named DISPLAY

Chapter 10

INTERACTIVE MEASUREMENTS

INTRODUCTION

This chapter describes the interactive measurement capabilities of the timing analyzer. It describes the way the timing analyzer uses the lines of the intermodule bus, and lists and defines the command syntax for setting up interactive measurements in the timing analyzer. Finally, two example measurements are described: one combining two timing analyzers, and the other using a timing analyzer and state analyzer together. Refer to the Measurement System Reference Manual for information about overall measurement system interaction.

GENERAL INFORMATION

Interactive measurements are coordinated measurements made with two or more analyzers. Interactive measurements are made through the intermodule bus cable connected to the IMB connectors on each analyzer control board. At power-up, there is no interactive specification between the timing analyzer(s) and any other analysis modules. In order to obtain interactive measurements, select the timing analyzer trace specification and enter the interactions desired.

You can have the timing analyzer receive enable levels or triggers from other analyzers in the system. You can also have the timing analyzer drive these enable levels and triggers to other analysis modules on the intermodule bus. Additionally, the timing analyzer can recieve a clock signal from the other analysis modules on the intermodule bus.

INTERMODULE SIGNALS

The timing analyzer can participate in intermodule activity on four of the intermodule control lines. The signals of the intermodule control lines are as follows:

MASTER ENABLE

The master enable line is shared by all analysis instruments included in a measurement. When master enable is true, it enables all instruments that receive it. When it is false, it freezes all these same instruments. There can be only one driver for the master enable line: either the *execute/halt* key or a subsystem, other than timing, which has been designated to participate in the measurement.

The primary purpose of the master enable line is to synchronize measurement start in all analyzers. At measurement start (the *execute* key pressed or true state from designated master enable driver), all analyzers try to start. As each analyzer gets ready to start, it allows the master enable line to go true, but the master enable line remains false until the last analyzer trys to start. Then the master enable line switches true, releasing all analyzers to start together.

When the master enable line is driven by one of the analyzers, it can alternate between true and false to window the activity in the interactive measurement for subystems other than timing. The timing analysis subsystem will use master enable to window only its trigger recognition activity. This results from the need to display continuous time sequences of data with the timing subsystem, so that its storage activity cannot be windowed. The timing analyzer cannot drive master enable.

TRIGGER ENABLE

When the trigger enable line is true, it enables the receiving instruments to recognize their triggers, if they occur. When it is false, it disables trigger recognition in the receiving subsytems. The trigger enable line can alternate between true and false during a measurement to allow the controlling analyzer to window the program activity in other analyzers where trigger recognition can occur. The timing analyzer can drive this line, or receive it from some other instrument on the intermodule bus. If no analyzer is designated to drive this line, it defaults to a true condition.

TRIGGER

The trigger line is normally false and gets latched true by the subsystem driving it. The trigger transition can be generated by the timing analyzer when it recognizes its trigger specification. The timing analyzer can also receive the trigger signal from another analyzer connected to the intermodule bus. In this case, the timing analyzer will identify the condition at its input as the trigger condition in its trace memory and on the trace list display. The trigger line can have more than one driver designated. In this case, the triggers are logically ORed together and the first driver to issue the trigger transition becomes the trigger source for the measurement in process. If no driver is specified, the trigger line defaults to false.

STORAGE ENABLE

The timing analyzer neither drives nor recieves this line since its data storage is uninterruptable. Otherwise, this line is used to control when the receiving subsystems are allowed to store information, storing states only when this line is true. The storage enable line can alternate between true and false during a measurement to window the activity to be stored. If no analyzer is designated to drive this line, it defaults to true.

DELAY CLOCK

The delay clock line carries a series of transitions issued each time an analyzer recognizes its incoming clocks. The timing analyzer can only receive clocks from this line. Other analyzers (such as state analyzers) can transmit these clocks and the timing analyzer can use them to delay its trace point by a selectable number of clocks.

TIMING ANALYZER SYNTAX FOR INTERACTIVE MEASUREMENTS

All of the selections affecting interactive measurements for the timing analyzer are made in the trace specification. The trace specification softkeys which make these selections are as follows:

TRIGGER ENABLE

1. **trigger enable received**. The timing analyzer recieves its trigger enable level from some other analyzer on the intermodule bus.

2. **trigger enable driven on_trigger**. The timing analyzer is always able to search for its own trigger, but until it finds it, the other subsystems receiving the enable level are not. When the timing analyzer satisfies its specification, it places the enable level on the intermodule bus, allowing the other subsystems to search for their triggers.

TRIGGER

NOTE

> When the timing analyzer trigger is *received*, *received_or_driven*, or *driven_on*, the "t" which indicates the trigger position on the display is replaced by an "i". The "i" indicates the point on the timing diagram when the IMB trigger line switched to true.

1. **trigger received**. The timing analyzer recognizes trigger when it receives the trigger transition on the intermodule bus. This point is independent of any input data pattern. It will be identified as the trigger point in the trace memory and timing display.

2. **trigger received_or_driven on <PATTERN> or <DELAYED PATTERN>**. The timing analyzer recognizes trigger either when it receives the transition from the trigger line of the intermodule bus, or when it satisfies its own internal specification. If it satisfies its own internal specification first, it will generate the trigger transition on the intermodule bus.

3. **trigger driven_on <PATTERN>** or **<DELAYED PATTERN>**. The timing analyzer will recognize the occurrence you specify as the trigger event. When it recognizes its trigger event, the timing analyzer will drive the trigger line in the intermodule bus.

TRIGGER ENABLE RESTART

trigger enable restart. This mode requires the timing analyzer and state/software analyzer to operate together on the IMB, and **trigger enable received** and **trigger enable restart** both to be specified. While the timing analyzer is performing a measurement, it monitors Low Trigger Enable (LTE). If LTE switches false during the measurement, the timing analyzer will purge its data memory and return to its prestart condition. It remains in the prestart condition until LTE switches true, again. Then the timing analyzer starts performing a new measurement according to its measurement setup (begins trigger search, etc). If the timing analyzer completes its measurement, it also checks High Trigger (HTR). If HTR is true, the timing analyzer will hold its data and display the trace results. If LTE switches false before HTR is true, the timing analyzer will again purge its data memory and return to the prestart condition.

EXAMPLE: MAKING INTERACTIVE TESTS WITH TWO OR MORE TIMING ANALYZERS

The following paragraphs describe how to set up two timing analyzers to make an interactive measurement.

1. From the monitor level of softkeys, press the *meas_sys* softkey and the **RETURN** key. This will gain access to the measurement system display. Add *continue* to your command if you want the timing analyzers to continue using their former test setups. The softkey label line will identify both timing analyzers by name and by the slot numbers where their control cards are installed.

2. Your identifiers will be similar to those shown in the following example of a softkey label line:

timing_1 *timing_4* ________ ________ ________ *end*

3. Press one of the *timing* softkeys (such as *timing_1*) to select the timing analyzer you want to set up first. Add the file name if you want to load a configuration from file memory. Then press **RETURN**. The trace specification of the timing analyzer whose control card is installed in the selected mainframe slot (slot 1 in this example) will appear on screen.

4. Make any desired modifications to the configuration. Add the desired interaction for this timing analyzer. Then press the *end* softkey and the **RETURN** key. The CRT will again show the measurement system display and will identify the interaction you assigned to your timing analyzer.

5. Press the other *timing* softkey (such as *timing_4*). Again, decide whether you want to use the default test setup or load a configuration from memory (by adding the name of the file). Press the **RETURN** key. This activates the other timing analysis module and brings its trace specification to the screen.

6. Make any desired changes to the measurement setup for this timing analyzer. Add the desired interaction to the specification.

7. Press the *end* softkey and the **RETURN** key to return to the measurement system level of display. This display will now show the interactions you specified for both timing analyzers.

NOTE

> You can *execute* or *halt* interactive measurements while the system displays the measurement system monitor or while it displays specifications from any one of the analyzers that will be included in the measurement.

8. Press the *execute* softkey and the **RETURN** key. This starts both timing analyzers running according to their own measurement configurations.

NOTE

> You can select any of the other analyzers in your mainframe and look at their present specifications, but you cannot execute any measurements with them as long as your interactive measurement is in process.

9. Press the *end* softkey again. This returns you to the monitor level of display. You can use the mainframe for nonanalysis activity (edit, assembly, etc) while the analyzers execute their measurements. Both timing analyzers will continue to run their measurements according to their specifications and save data in their memories.

10. To look at the results of the timing analysis measurements, press the *meas_sys* and *continue* softkeys, then press the **RETURN** key. This command line returns the system to the measurement system display without changing the analyzer setups.

11. Now press the *timing<number>* softkey and the **RETURN** key. This regains access to the timing analyzer you select without reloading the module. The measurement you assigned to the timing analyzer will still be running if not already completed. You can select displays and use the capabilities of the timing analyzer as though it were the only analyzer in the mainframe.

12. When you have finished with that timing analyzer, press the *end* softkey and the **RETURN** key.

13. You can check the status of the measurement in the other timing analyzer. Press the other *timing<number>* softkey and the **RETURN** key.

EXAMPLE: USING A TIMING ANALYZER AND STATE ANALYZER TOGETHER

Assume the system under test executes normal software within the display-update and keyboard-read routines. Occasionally, it jumps outside those routines to another area of software and the display fails.

You would like to know what causes the system to exit its normal routines, and where it goes after the error condition occurs.

STATE ANALYZER SETUP

You can set up the state analyzer to trigger a trace when it detects any address outside the range of normal routines and at the same time, trigger a timing measurement in the associated timing analyzer. Use a setup such as:

trigger int/drive on ADDRESS <> range DISPLAY

or ADDRESS <> range KEY_INPUT **RETURN**

store on any_state **RETURN**

The above specification combines the range of addresses involved with display update and the range of addresses involved with keyboard reading into one range. For the state analyzer, these two ranges need not be contiguous.

TIMING ANALYZER SETUP

Set up your timing analyzer to receive its trigger from the intermodule bus and to position that trigger near the middle of its memory. Use a setup such as:

trigger received **RETURN**

trigger position center **RETURN**

ANALYSIS EXECUTION

Press the *execute* softkey, then the RETURN key to start your measurement. When your software under test leaves the range of the normal routines, the state analyzer will trigger a trace. It will also trigger a measurement in the timing analyzer.

The state analyzer trace list will show which addresses were being executed when the error condition occurred, and which new address range was entered after the error condition. This may not be enough to identify the cause of the error. The timing analyzer may show a glitch on one of the control lines just before the trigger point. The system could be reading the glitch as a jump instruction.

By making a series of measurements, you can determine which components of the error activity are consistent and which components seem to happen at random intervals. This information should lead to the cause of the problem.

NOTE

> If trigger is driven or received from the IMB, then the trigger location (usually marked "t" on the display) is marked with "i" instead, to indicate the point in memory where the trigger signal occurred on the intermodule bus.
>
> If using "trigger driven on ...", note that the trigger condition occurred 50 ns before the trigger signal was placed on the bus. This will cause the "i" indicator to be placed 50 nsec later than the true trigger point on the display.

Chapter 11

THEORY OF OPERATION

INTRODUCTION

This chapter contains the theory of operation for the timing analyzer. This theory of operation is based on operator requirements rather than troubleshooting. For the theory of operation pertaining to troubleshooting, refer to the timing analyzer service manual.

GENERAL THEORY

The theory of operation contained in this section discusses asynchronous sampling and asynchronous triggering, time interval resolution, how to determine the accuracy of interval measurements, standard-deviation measurements, and mean value measurements, and sources of errors in sampled data.

ASYNCHRONOUS SAMPLING

The timing analyzer samples data asynchronously from the system under test. That is, the sample clock (which is internal) is not synchronized with the state flow in the system under test. The sample clock is periodic and may be used for time measurement. The states on the input data lines are captured when each sample clock occurs until the entire memory is filled with sequential sampled states.

ASYNCHRONOUS TRIGGERING

In the timing analyzer, the trigger function is not synchronous with the sampling clock. The trigger function operates separately to enable trigger recognition on narrow phenomena which would not likely be sampled, especially when using a slow sampling clock rate. The trigger position is marked on screen by the timing analyzer even though the trigger event may not have been sampled.

Trigger states must be present at the probe inputs for 6 ns (typically) to be recognized, except for glitch triggers used in the glitch capture mode. In the glitch capture mode, the minimum width (3 ns) is set by the glitch capture circuitry.

TIME INTERVAL RESOLUTION

Time interval measurement resolution is one of the most important specifications of a timing analyzer. The resolution depends on three factors:

1. Sample period.

2. Interchannel skew.

3. Memory depth.

The resolution when measuring a time interval is determined as follows:

Resolution = plus or minus (sample period + skew)

Skew is differences in delays of the probe channels: delay differences from one channel to another, and delay differences in recognizing negative and positive transitions.

The high sample rate (400 MHz) coupled with low skew (plus or minus 1.5 ns for both opposite and same direction transitions) give the timing analyzer very good resolution.

Skew, by itself, is a function of several input variables, listed as follows:

1. Input signal slew rate in volts per nanosecond (a low slew rate increases skew).

2. Signal overdrive above the threshold, as a percent of amplitude (low overdrive increases skew).

3. Threshold value selected (high threshold settings increase skew).

The skew specification of 1.5 ns for all probes within one pod is measured under the following conditions:

1. A 0.25-volt per nanosecond slew rate.

2. A 0.6-volt amplitude signal with equal swings on either side of threshold.

3. A minus 1.3-volt threshold.

Memory depth is also important in time interval resolution. It sets the maximum time interval that can be measured with any sample period. In the 400-MHz (fast sample) mode of operation, a time interval of 20.4 usec can be measured with full sample accuracy, i.e.; (8140 samples of memory depth) (2.5 ns per sample) = 20.4 usec. To measure longer time intervals, the sample period must be increased.

ACCURACY OF INTERVAL MEASUREMENTS

You can improve the accuracy of an interval measurement by making the measurement with a series of repetitive executions. When a single execution is made, the measurement accuracy is equal to +/- the sample period + skew. When measuring a stable interval using a series of repetitive executions, the accuracy of the measurement improves by the following formula:

accuracy = +/- (sample period)/sqrt(n) + skew

where: sqrt(n) = the square root of the number of executions included in the measurement.
sample period = the sample period specified in the timing analyzer trace specification.
skew = the delay differences between input channels.

The time interval being measured must not be synchronous to the sampling clock of the timing analyzer.

Using the above formula, an improvement of X10 is obtained in the accuracy of your measurement by using 100 repetitive executions to obtain your measurement.

ACCURACY OF MEAN VALUE MEASUREMENTS

The accuracy of the displayed mean value of a single inverval depends on the number of executions in the series used to determine mean value. Assume the timing analyzer is measuring a stable, repetitive time interval approximately 100 usec long. Using the 20-MHz sample rate, this sample rate, you capture 203 usec of timing data because:

4060 * 50 ns = 203 usec

where: 4060 = the depth of the timing analyzer memory.
50 ns = the selected sample period.

A single measurement will have the following accuracy:

+/-(50 ns + 1.5 ns) = +/- 51.5 ns

where: 1.5 ns = the skew specification for a single pod.
50 = is the sample period.

By making 100 measurements in a repetitive series, the accuracy of the mean value displayed will be improved by a factor of the square root of the number of traces included in the series.:

+/-[(50 ns/sqrt(100) + 1.5 ns] = +/- 6.5 ns

ACCURACY OF STANDARD-DEVIATION MEASUREMENTS

An interval that does not vary can still be shown to have a large standard deviation due to the sampling process. The error in the displayed standard deviation depends on the size of two elements: (1) the portion of the interval that exceeds the multiple of the sample periods, and (2) the portion of the interval that includes complete sample periods.

Example 1: Assume the timing analyzer is measuring a time interval of exactly 12.5 ns. It makes 10 executions using a 5-ns sample period. Five of the executions show the interval to last 15 ns (3 sample clocks), and five of the executions show the interval to last 10 ns (2 sample clocks). Even though the input signal has a true standard deviation of 0.0 ns, the timing analyzer will calculate the standard deviation of this signal to be 2.64 ns and display this standard deviation on screen. When the sampled standard deviation is less than one sample period, its value is mainly determined by the sampling process.

Example 2: Assume the timing analyzer is measuring a time interval that varies from 5 to 8 usec. The timing analyzer is operating with a 5-ns sample period. After a series of repetitive measurements, the timing analyzer shows a standard deviation of 1.5 usec for the interval being measured. This dispersion is determined by variations in the time interval itself, and not the sampling process. In this case, the standard deviation is much larger than one sample period.

ERROR SOURCES

ERRORS CAUSED BY DATA AND SAMPLED CLOCK BEING SYNCHRONOUS

When making a trace of data that is synchronous to the sampling clock, the data will be sampled at the same relative point throughout the trace. Multiple measurements may not necessarily improve accuracy. Still, the probability that this would degrade a large number of measurement executions is small because of the way the timing analyzer performs multiple executions. The timing analyzer executes a trace and then halts while it marks the time interval in question, updates its statistical information, and presents the display. Then it begins a new trace. The time between consecutive executions will be approximately 1 second. This makes it unlikely that the position of the data and sampling clock will be the same during the next execution.

STATISTICAL ERRORS CAUSED BY SAMPLING PROCESS

The timing analyzer calculates statistics on the sampled data in its memory, not on all of the data generated by the system under test. The data in memory may misrepresent the actual data. Misleading data can be captured when you trigger your trace on some occurrence that causes the timing analyzer to capture samples at misleading points in the data flow of the system under test.

Use of the "trigger on any_state" specification may not overcome all measurement bias problems. Consider the case where the timing analyzer is measuring an interval of time between positive edges occurring on a probe line. Suppose there are two intervals on that line, and they are occurring alternately: one is 10 usec long and the other is 20 usec long. Interval measurements are made by marking "x" on the first positive edge of the selected label and marking "o" on the next positive edge after "x". The random beginning of a new trace will probably occur twice as often during the 20-usec interval as during the 10-usec interval. Because of this, the timing analyzer will appear to be finding twice as many 10-usec intervals as 20-usec intervals, but in the system under test, there are equal numbers of 10-usec and 20-usec intervals.

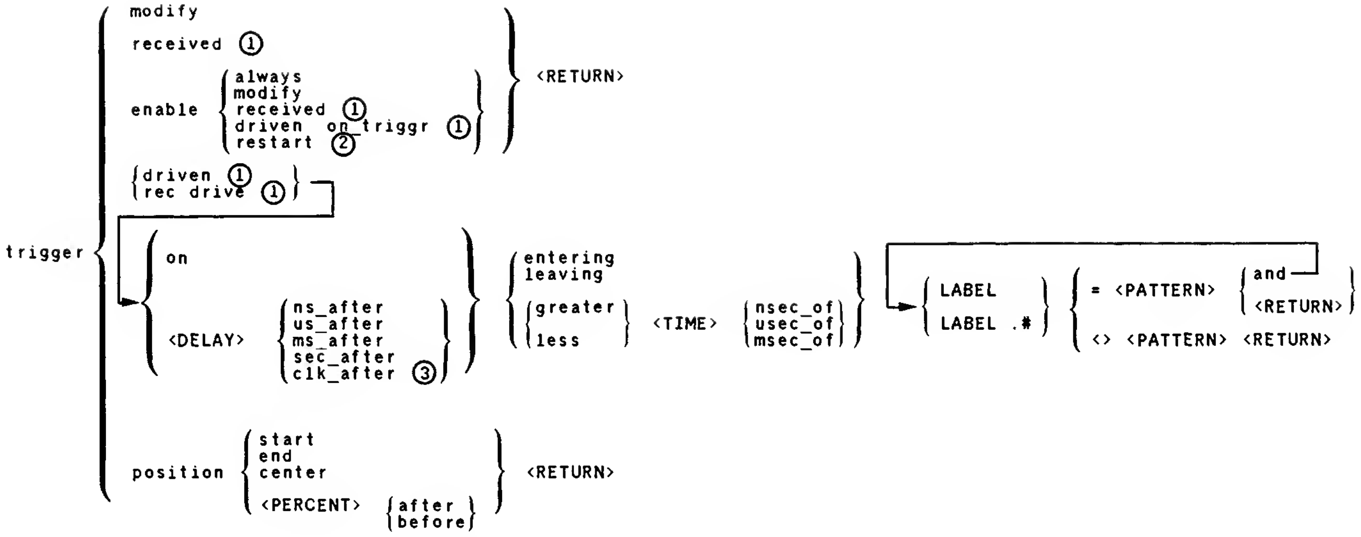
One possible approach to solving the problem of misleading data in the above example is to find another line with a uniform square wave operating at twice the frequency of the combined intervals. Such a square wave will have as many positive edges preceding 10-usec intervals as 20-usec intervals. By triggering the interval measurements on positive edges in that square wave, and marking "x" and "o" on the first interval after each trigger, the timing analyzer will measure as many 20-usec intervals as 10-usec intervals.

Appendix A

OPERATING SYNTAX

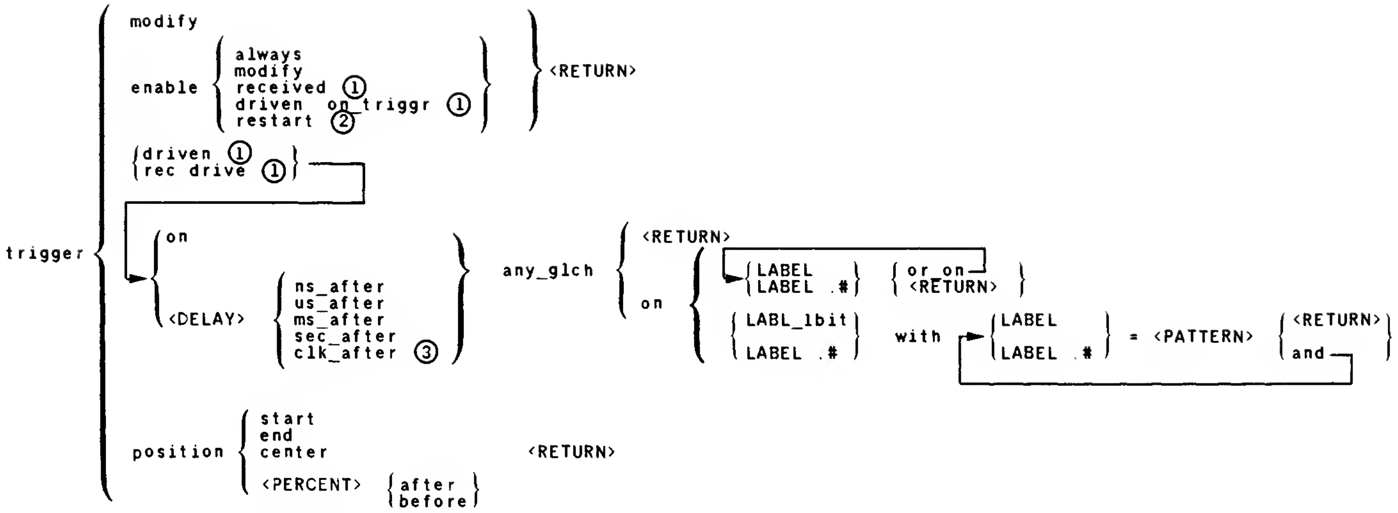
INTRODUCTION

This appendix contains the *trigger*, *threshold*, *define*, *display*, *copy*, *magnify*, *mark*, *find*, *process_for_data*, and *halt_repetitive_execution* operating syntax.



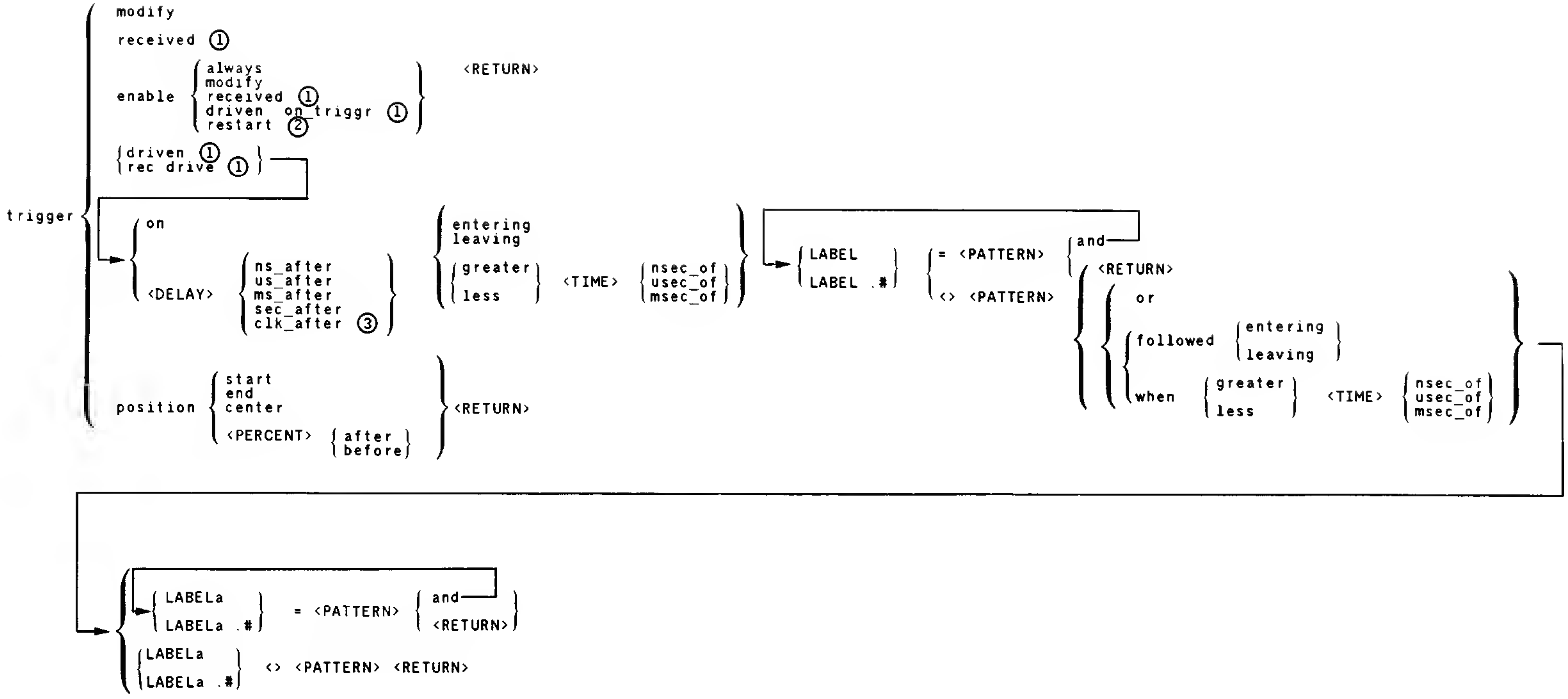
NOTES
1 Requires an additional module in the frame
2 Requires a State Analysis Module in the mainframe and `trigger enable received'
3 Requires a State or an Emulation External Analysis Module in the mainframe
4 <DELAY> is an integer in the range 1 to 32767, and in time mode the range is from 2 sample periods to 32 million sample periods
5 <TIME> is an integer in the range 1 to 32767 with allowed values in a 1-2-5 sequence, with a time range from 5nS to 1mS for greater than triggers and from 10nS to 1mS for less than triggers; values not entered in the 1-2-5 sequence are rounded up
6 <PERCENT> is an integer in the range 0 to 100 rounded to the nearest multiple of 5
7 # is the desired single bit location within the label, counting from 0 on the right

Figure A-1. 8-Channel Asynchronous Trigger Syntax



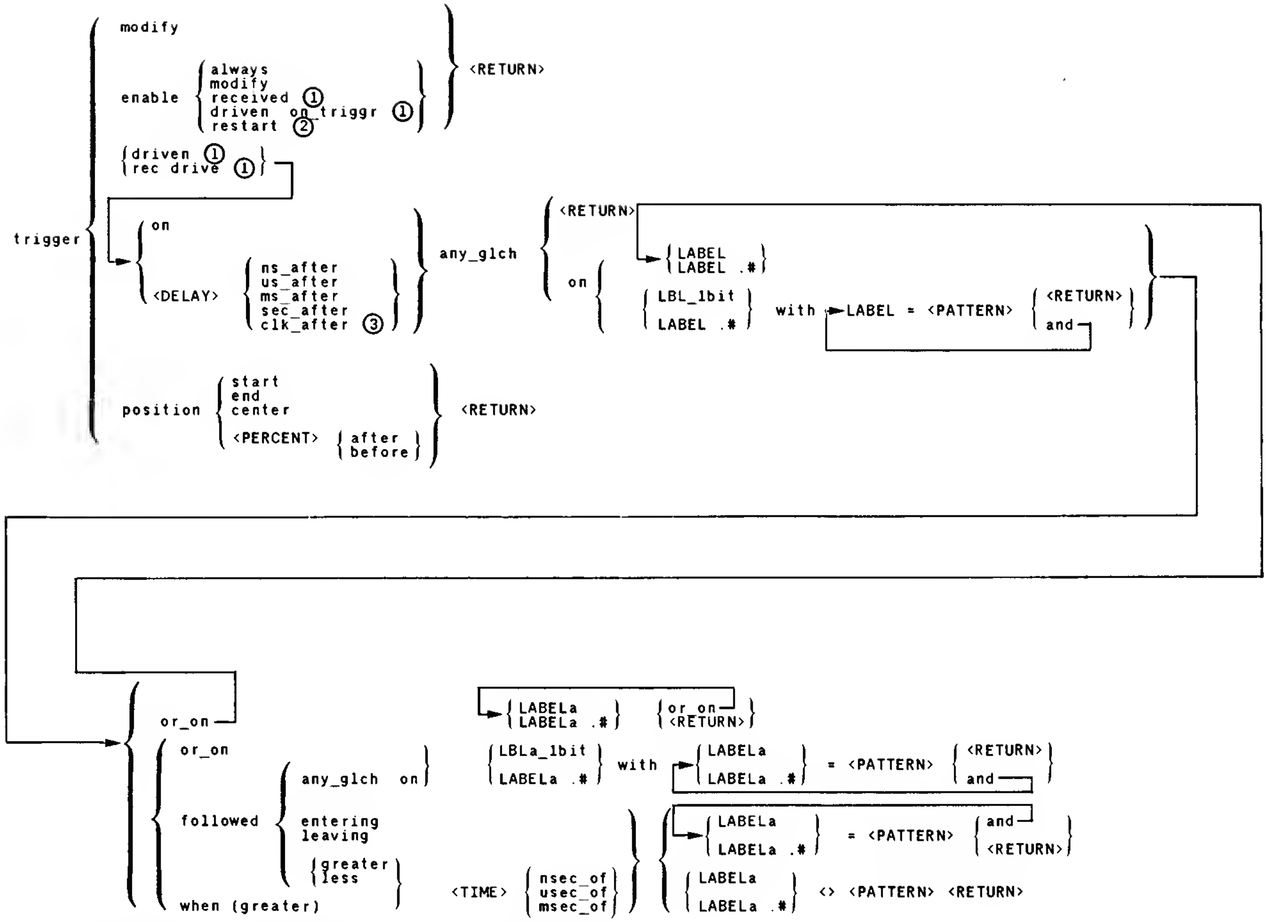
NOTES
① Requires an additional module in the frame
② Requires a State Analysis Module in the mainframe and `trigger enable received'
③ Requires a State or an Emulation External Analysis Module in the mainframe
4 <DELAY> is an integer in the range 1 to 32767, and in time mode the range is from 2 sample periods to 32 million sample periods
5 <TIME> is an integer in the range 1 to 32767 with allowed values in a 1-2-5 sequence, with a time range from 2 sample periods to 1mS; values not entered in the 1-2-5 sequence are rounded up
6 <PERCENT> is an integer in the range 0 to 100 rounded to the nearest multiple of 5
7 # is the desired single bit location within the label, counting from 0 on the right
8 LABL_1bit is a label defined on only one bit

Figure A-2. 8-Channel Sampled Data Trigger Syntax



NOTES
① Requires an additional module in the frame
② Requires a State Analysis Module in the mainframe and `trigger enable received'
③ Requires a State or an Emulation External Analysis Module in the mainframe
4 LABELa or LABELa .# is in a different pod than LABEL or LABEL .#
5 <DELAY> is an integer in the range 1 to 32767, and in time mode the range is from 2 sample periods to 32 million sample periods
6 <TIME> is an integer in the range 1 to 32767 with allowed values in a 1-2-5 sequence, with a time range from 5nS to 1mS for greater triggers and from 10nS to 1mS for less than triggers; values not entered in the 1-2-5 sequence are rounded up
7 <PERCENT> is an integer in the range 0 to 100 rounded to the nearest multiple of 5
8 # is the desired single bit location within the label, counting from 0 on the right

Figure A-3. 16-Channel Asynchronous Trigger Syntax



NOTES
① Requires an additional module in the frame
② Requires a State Analysis Module in the mainframe and `trigger enable received'
③ Requires a State or an Emulation External Analysis Module in the mainframe
4 LABELa or LABELa .# is in a different pod than LABEL or LABEL .#
5 <DELAY> is an integer in the range 1 to 32767, and in time mode the range is from 2 sample periods to 32 million sample periods
6 <TIME> is an integer in the range 1 to 32767 with allowed values in a 1-2-5 sequence, with a time range from 2 sample periods to 1mS; values not entered in the 1-2-5 sequence are rounded up
7 <PERCENT> is an integer in the range 0 to 100 rounded to the nearest multiple of 5
8 # is the desired single bit location within the label, counting from 0 on the right
9 LABL_1bit is a label defined on only one bit

Figure A-4. 16-Channel Sampled Data Trigger Syntax

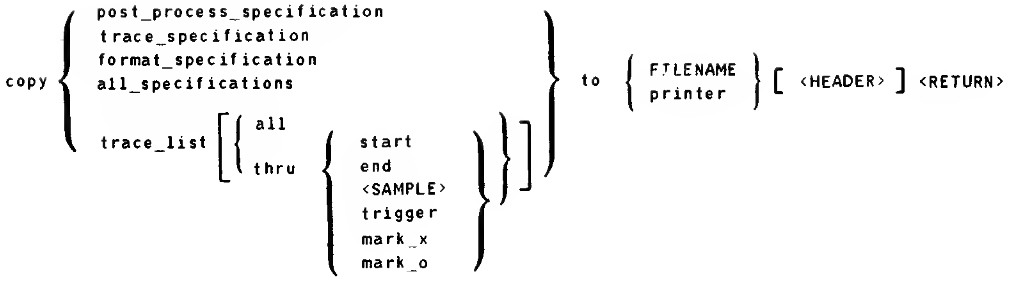
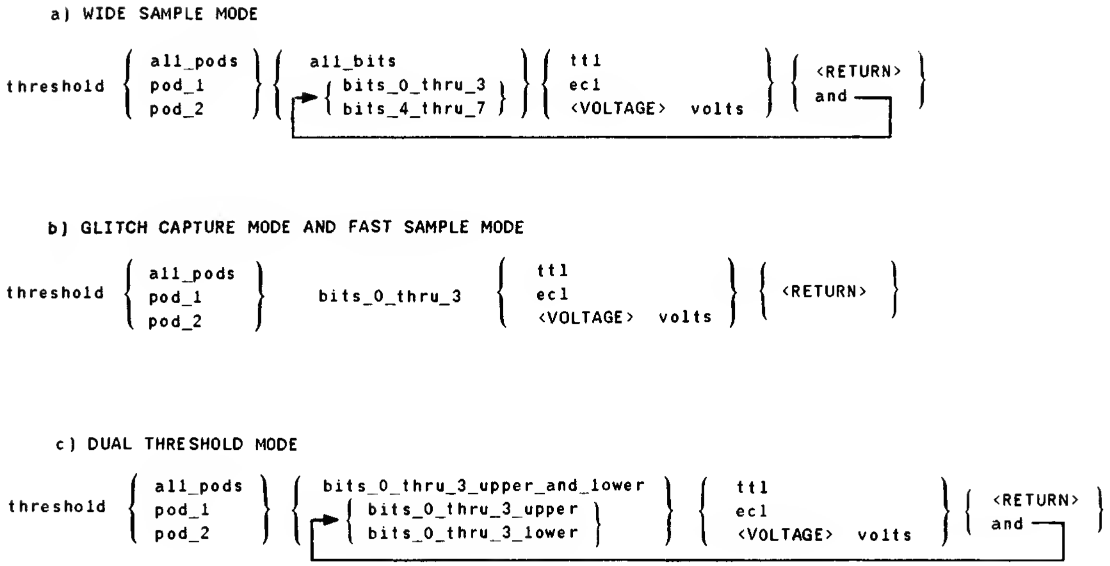


Figure A-5. Copy Syntax



where <VOLTAGE> is any number between -10 & + 10 in 0.1 volt steps

Figure A-6. Threshold Syntax

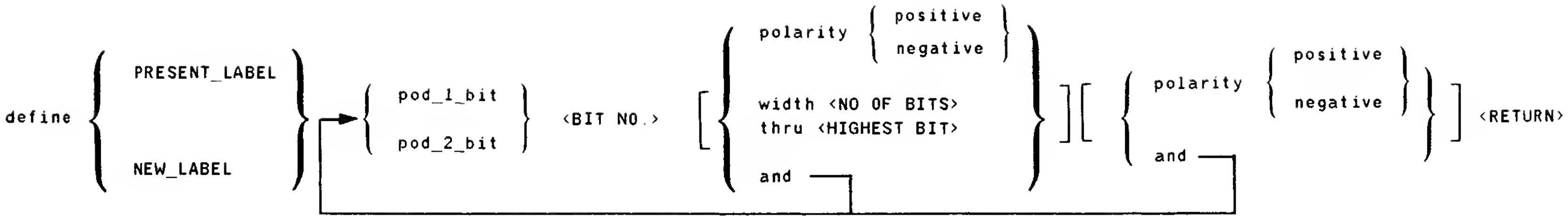


Figure A-7. Define Syntax

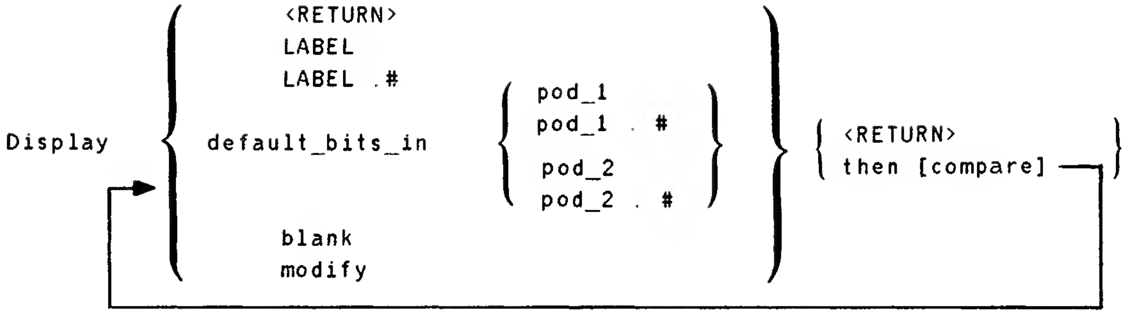


Figure A-8. Display Syntax

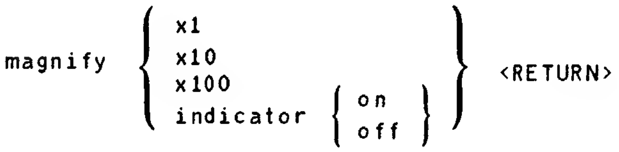


Figure A-9. Magnify Syntax

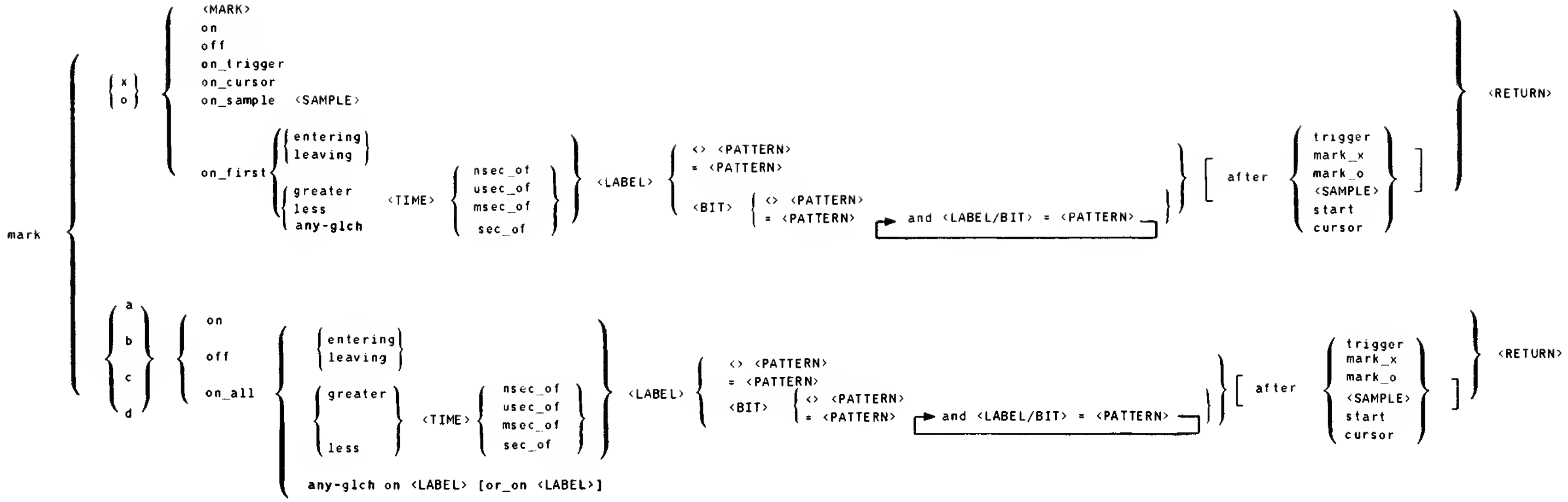


Figure A-10. Mark Syntax

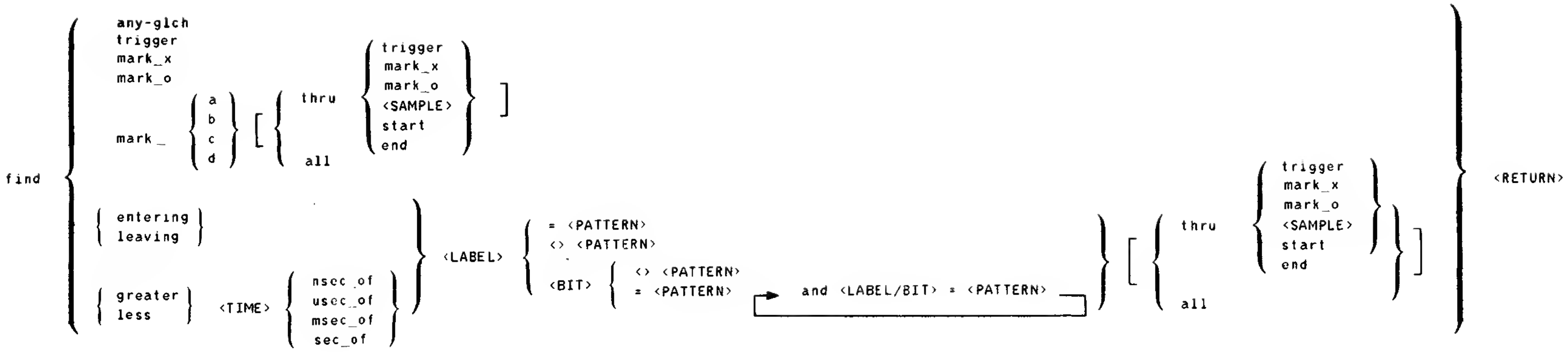


Figure A-11. Find Syntax

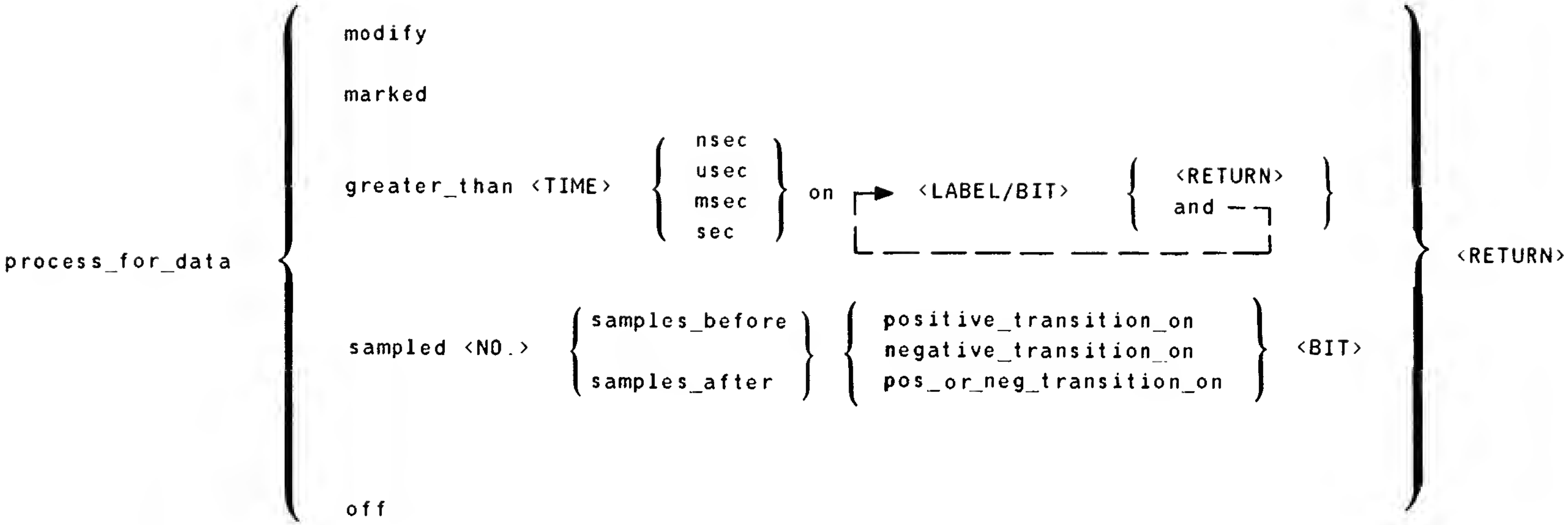


Figure A-12. Process__For__Data Syntax

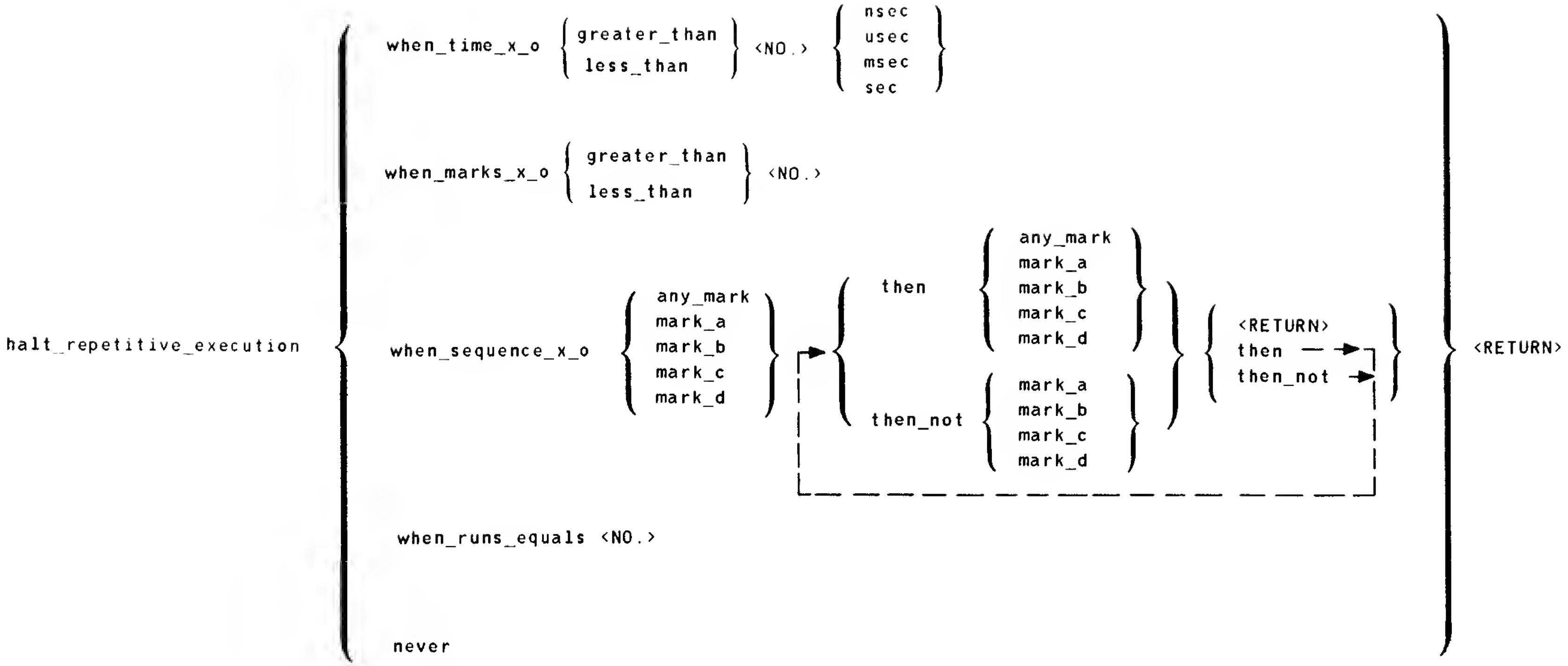


Figure A-13. Halt__Repetitive__Execution Syntax

Appendix B

GLOSSARY OF SPECIFICATION SOFTKEYS

INTRODUCTION

The following glossary contains all of the keywords which appear on the softkeys and in commands in the timing analyzer. The display areas where each keyword appears are indicated by the following abbreviations:

TS = Trace Specification
FS = Format Specification
PP = Post_process Specification
TD = Timing Diagram Display
TL = Trace List Display

absolute

absolute
Used with *display* command to define that the time displayed will be total time between the sample on each line and the time when the trigger occurred.
TL - *display POD1 then time__count absolute*

after

after
Used with the *mark* command to define a range of memory where the identifying mark(s) is to be assigned.
TD, TL, PP - *mark__x on__first__occurrence__of entering POD1 .6 = 1 after trigger*

all

all
Used with the *copy trace__list* command sequence to define that the entire list is to be copied to the destination file or device.
TL - *copy trace__list all to printer*
TL - *find mark__a all*

all__bits

all__bits
Used with *threshold* command to define that this threshold voltage specification should apply to bits 0..7 in the pod being specified.
FS - *threshold pod__1 all__bits ecl*

all__labels

all__label
Used with the *delete* command to erase all label definitions. Only labels not used in any mode Trace Specification will be deleted.
FS - *delete all__labels*

all_pods *all_pods*

Used with the *threshold* command in a 16-channel analyzer to define that this threshold voltage specification applies to both pod 1 and pod 2.

FS - *threshold all_pods all_bits ecl*

all_specifications *all_specs*

Used with the *copy* command to define that the Trace and Format Specifications and Trace List are to be copied to the destination file or device. Used with the *default* command to restore the Trace, Format, and Post_process Specifications, and the Trace List and Timing Diagram display formats to their default conditions and labels.

TS, FS, PP, TD, TL - *copy all_specifications to printer*

TS, FS, PP - *default all_specifications*

always *always*

Used with the *trigger enable* sequence to reset any interactive trigger specification so as not to listen to the Low Trigger Enable line in the Inter Module Bus. This will clear any trigger enable out of the specification, causing it to disappear from the display.

TS - *trigger enable always*

and *and*

Used with the *trigger* command to include additional labels in the trigger condition being defined. Used with the *threshold* command in Wide Sample Mode when upper and lower pod inputs are to be compared to different voltage values, and in Dual Threshold Mode to set the second threshold. Used with the *define* command to extend the definition to additional inputs.

TS - *trigger on entering POD1. 0 = 0 and POD1. 1 = 1*

FS - *threshold pod_1 bits_0_thru_3 ecl and pod_1 bits_4_thru_7 ttl*

FS - *define LABEL pod_1_bit 0 and pod_1_bit 1*

any_glitch *any_glch*

Used with the *trigger* command during Glitch Capture Mode to include a glitch or multiple glitches in the trigger definition. Used with the *find* command during Glitch Capture Mode to find a sample where a glitch or multiple glitches were detected during the trace measurement.

TS - *trigger on any_glitch on POD1lower .0*

TD, TL - *find any_glitch on POD1lower .1 or_on POD1lower .2*

TD, TL __ *mark o on_first_occurrence_of any_glitch*

any_mark *any_mark*

Used with the *halt_repetitive_execution* command to designate a don't-care member of a sequence of occurrences which, when found in order, will cause the analyzer to halt the repetitive executions in progress.

PP - *halt_repetitive_executions when_sequence_x_o mark_a then mark_a then any_mark then_not mark_c* *any__thing* *any__thing*

Used with the *trigger* command to establish a "don't care" trigger condition which will be satisfied by any probe input pattern.

TS - *trigger on any__thing*

assert *assert*

Command used to turn on or off the trigger output to the rear panel BNC connector.

TS - *assert bnc__port__4 on__internal__nondelayed__trigger*

blank *blank*

Used with the *display* command to produce an unlabeled null trace in the timing diagram, typically used between traces for better readability.

TD - *display POD1. 0 then blank then POD1. 1*

bits__0__thru__3 *bits__0__3*

Used with the *threshold* command in all modes except Dual Threshold to set the voltage specification against which probe pod inputs 0..3 will be compared.

FS - *threshold pod__1 bits__0__thru__3 ttl*

bits__0__thru__3__lower *lower*

Used with the *threshold* command in Dual Threshold Mode to set the voltage specification for the more negative threshold voltage against which to compare inputs 0..3.

FS - *threshold pod__1 bits__0__thru__3__lower 0.8 volts*
and bits__0__thru__3 upper 2.0 volts

bits__0__thru__3__upper *upper*

Used with the *threshold* command in Dual Threshold Mode to set the voltage specification for the more postive threshold voltage against which to compare inputs 0..3.

FS - *threshold pod__1 bits__0__thru__3__lower 0.8 volts*
and bits__0__thru__3 upper 2.0 volts

bits__0__thru__3__upper__and__lower *u__and__l*

Used with the *threshold* command in Dual Threshold Mode to set the voltage specification for both threshold voltages against which to compare inputs 0..3 to either ttl or ecl. These set the upper and lower thresholds to +2.0V & +0.8V, and -1.1V and -1.5V, for ttl and ecl, respectively.

FS - *threshold pod__1 bits__0__thru__3__upper__and__lower ttl*

bits__4__thru__7 *bits__4__7*

Used with the *threshold* command in Wide Sample Mode to set the voltage specification against which probe pod inputs 4..7 will be compared.

FS - *threshold pod__1 bits__4__thru__7 ttl* *bnc__port__4* *bnc__4*

Used with the *assert* command to define the conditions for driving rear panel BNC connector #4.

TS - *assert bnc__port__4 on__internal__nondelayed__trigger*

center__of__trace *center*

Used with the *trigger position__is* command to locate the relative position of the trigger in the center of the acquisition memory, and in X1 magnification, to the center of the displayed trace.

TS - *trigger position__is center__of__trace*

compare__file *compare*

Used with the *display* command to set up a display which includes information from your compare file.

TD, TL - *display POD1 .6 then compare__file POD1 .6*

compare__file__is *compare*

Command used to identify a configuration that was stored with data to be accessed by the Timing Analyzer and used as a reference source when you are making comparison measurements, or when cancelling an existing compare file specification.

PP - *compare__file__is FILE4*

configuration *configure*

Command used to create a file of type trace which contains the current Trace, Format, and Post_process Specifications and Display formats for future recall, along with the content of the trace memory (if desired), or to load such a file to set up these specifications again. Should the current timing analyzer board assembly configuration not be identical to that from which the file was created, labels which will be invalid are shown as null entries in the Format Specification and on execution, an error message will prompt for invalid Trace Specification entries.

TS, FS, TD, TL, PP - *configuration save__in FILENAME:USERID*

copy *copy*

Command used to transfer the Trace, Format, Post_process Specifications, Trace List, and/or Timing Diagram to a destination file or device.

TS, FS, PP, TD, TL - *copy all__specifications to printer* *cursor* *cursor*

Command used to cause the <-- and --> keys to operate on the vertical cursor in the diagram rather than on the flashing underline cursor on the command line. The *cursor* softkey is shown in inverse video while this mode is in effect. It can be cancelled by pressing the *cursor* softkey again. Because the command line cursor is eliminated, keyboard entries are inhibited. This command operation is analogous to the revise and insert commands in the Editor.

TD, TL - *cursor*

default *default*

Command used to return specifications to their preset conditions. Defaults are:

(Trace) sample rate *200MHz*;
trigger on *anything*;
(Format) define *POD1*, *POD1upper*, *POD1lower*,
POD2, *POD2upper*, *POD2lower*;
threshold *all_pods all_bits ttl*;
(Diagram) display *POD1 then POD2 then default_bits_in pod_1*
then default_bits_in pod_2 ;
(Trace List) display *POD1 in_binary then POD2 in_binary then*
time_count absolute .

TS, FS, PP - *default all_specifications*
TS, FS, PP - *default trace_specification*

default_bits_in *default*

Used with the *display* command to produce traces labeled as pod_#_bit_#.

TD - *display POD1 then default_bits_in pod_1*

define *define*

Command used to associate a new label name with probe inputs and a logic polarity. Up to 31 labels can be defined in the Format Specification. Default labels for a 16-channel analyzer are POD1, POD1upper, POD1lower, POD2, POD2upper, and POD2lower.

FS - *define NEWLABEL pod_1_bit 1 logic_polarity*
positive_true

delay_clocks_after *clk_after*

Used with the *trigger* command to delay the trigger position by an integral number of positive clock edges received on the Master Clock line of the Intermodule Bus, usually driven by the clock in the system under test as perceived by a synchronous analyzer module. A maximum delay of 32767 clocks is possible.

TS - *trigger 5 delay_clocks_after entering POD1 = 0*

delete *delete*

Command used to remove a label from the Format Specification. If this label is in use in any Specification in any of the four modes, the command will be ignored.

FS - *delete OLDLABEL*

diagram *diagram*

Command used to select either 8-channel or 16-channel vertical display format.

TD - *diagram eight__channels*

display *display*

Command used to set up the sequence of traces to be shown. A maximum of 31 traces can be set up in the Timing Diagram, although only 16 can be shown at one time. A maximum width of 132 characters can be set up in the Trace List, although only the first 80 characters will appear on screen. In the timing diagram, the display key used alone will switch between labels and probe bit identifiers for each trace on screen. The default for an eight-channel analyzer is:

TD - *display POD1 then default__bits__in POD1*
TL - *display POD1 then time__count absolute*

driven *driven*

Used with the *trigger* command to define that the timing analyzer trigger is to be output to other modules via the High Trigger or Low Trigger Enable lines in the Intermodule Bus.

TS - *trigger driven on POD1 = 0*
TS - *trigger enable driven on__trigger*

dual__threshold *dual*

Used with the *mode__is* command to set the operating mode of the analyzer to compare each of the 0..3 inputs on each probe pod to two different thresholds and display the result as a three-level trace. If the thresholds have been set previously in this mode, they will return to those values. Otherwise, if the thresholds were set to a voltage value in the previous mode, they will be shifted from that value(s) by +200 mV and -200 mV for the upper and lower thresholds, respectively; if they were set to ecl, they will go to -1.1V and -1.5V; if they were set to ttl or were not set, they will go to +2.0V and +0.8V.

TS, FS - *mode__is dual__threshold*

ecl *ecl*

Used with the *threshold* command to set the probe input comparison voltage to the middle of the 10K series ECL voltage range, -1.3V. In Dual Threshold Mode, the upper and lower thresholds will be set to -1.1V and -1.5V, respectively.

FS - *threshold pod__1 all__bits ecl*

eight__channels *8__chanls*

Used with the *diagram* command to set the vertical display format to eight traces.

TD - *diagram eight__channels*

enable *enable*

Used with the *trigger* command to qualify triggers for or from other modules via the Low Trigger Enable line on the Inter Module Bus. Note that Trigger Enable and Trigger cannot both be received.

TS - *trigger enable received*

end *end*

Command used to exit the Timing Analyzer sofware and return to the next higher level, either Measurement_System or system monitor, and to save a copy of the current configuration in a system-defined trace file to be used for automatic re-entry when transferring between modules in a multimodule measurement or when the coninue option is used to re-enter either *timing* or *meas_sys*. The trace file name will be Tdc#$:HP where # is the slot number of the Timing Control card and $ is the HP-IB address of the frame if in a cluster environment or 8 if in stand-alone mode. Used in the *copy* command to indicate the range of the copy is from the present cursor position to the end of the trace memory.

TS, FS, PP, TD, TL - *end*
TL - *copy trace_list thru end*

end_of_trace *end*

Used with the *trigger position_is* command to define the position of the trigger within the acquisition memory as 30 samples before the end of memory (70 in Fast Sample Mode).

TS - *trigger position_is end_of_trace*

entering *entering*

Used with the *trigger* command to define the trigger condition as the point where the trigger pattern comparison changes from not-true to true. The not-true condition must precede the true; if the condition is always true, triggering will not occur. If the trigger is output to the rear panel BNC connectors, the signal there will be a narrow pulse. Note that an ANDed condition will go true when the total condition goes true, e.g.- all elements are true but one and then that element goes true, causing the trigger. Used with the *find* and *mark* commands to define the point to be found (and/or marked) is the point where a specified pattern changes from not-true to true. The not-true condition must precede the true; if the condition is always true, it will not be found.

TS - *trigger on entering POD1. 0 = 0 and POD1. 1 = 1*
(note that *trigger on entering POD1 = 0XXH* will
never trigger)
TD, TL - *find entering POD1 .6 = 1 and POD1 .5 = 0*
thru mark_x
TD, TL, PP - *mark a on_all_occurrences_of entering*
POD1 .6 = 1 and POD1 .7 = 1

execute *execute*

Command to cause the timing analyzer to start capturing data. If any Trace Specifications have changed since the last measurement, the new setup will be loaded before capturing data. When the defined pretrigger data has been stored, the analyzer will then begin checking for its trigger condition which will be used to determine when the measurement is complete.

TS, FS, PP, TD, TL - *execute*

fast_sample *fast*

Used with the *mode_is* command to set the operating mode of the analyzer to sample each of the 0..3 inputs on each probe pod at a 400-MHz rate with a memory depth of 8140 samples.

TS, FS - *mode_is fast_threshold*

find *find*

Command to bring any specified event to the display, centered if possible, and locate the cursor at that point. Using this command, you can find the trigger, or any of the mark_x, o, a, b, c, or d samples by specifying the mark_identifiers. You can also find types of occurrences such as when a new value first enters or a present value first leaves in a probe channel, or when a condition lasts for a time greater than or less than a specified time.

TD, TL - *find mark_x*

followed_by *followed*

Used with the *trigger* command to establish a trigger sequence composed of two conditions. The second condition can be recognized only after the first condition has been satisfied. Note that by this definition, a second condition which is true when the first condition becomes true will produce a trigger. No ambiguity can occur if you use the *entering* or *leaving* constructs to define your second condition.

TS - *trigger on entering POD1 = 0 followed_by entering POD2 = 0*

format_specification *format*

Used with the *show* command to bring the Format Specification on the screen for setting labels, thresholds, and logic polarities.

TS, PP, TD, TL - *show format_specification* *glitch__capture* *glitch*

Used with *mode__is* command to set the operating mode of the analyzer to examine each of the 0..3 inputs on each probe pod for both data and glitches. A glitch is defined as multiple transitions between adjacent data sample times. Triggers involving glitches are synchronous with durations in multiples of sample periods. A Glitch is displayed at X100 magnification as adjacent broken vertical bars. At X10 and X1, a single broken bar is used to identify both glitches and multiple data transitions occurring too near each other to be displayed separately.

TS, FS - *mode__is glitch__capture*

greater__than *greater*

Used with the *trigger* command to time-qualify the trigger pattern, allowing for trigger recognition only on those patterns which exist for a time greater than the specification. The trigger position will indicate where the time specification was satisfied. If the trigger is output to the rear panel BNC connectors, the signal there will be that portion of the trigger pattern that exceeds the time specification. The available range is from 5 nsec to 1 msec for asynchronous triggers and from two sample periods min to 1 msec max for synchronous triggers. Used with the *process* command to qualify the display of only those samples which exceeded the time specification. Used with the *find* command to define the point to be found as the point where a specified pattern remains true for more than a particular time period. The analyzer can only find the specified condition within the range of memory you designate. Used with the *halt__repetitive__execution* command to designate a number of marks or a period of time which, when exceeded, will cause generation of the halt command for a repetitive execution in progress.

TS - *trigger on greater__than 50 nsec__of POD1 = 0*

TL, PP - *process__for__data greater__than 30 usec on POD1 .6*

TD, TL - *find greater__than 30 usec__of POD1 .6 = 1 and POD1 .5 = 0 thru mark__x*

PP - *halt__repetitive__execution when__marks__x__o greater__than 40*

halt *halt*

Command available only while executing, to stop the current execution. If memory has been filled, the entire memory can be displayed, otherwise only that portion of memory just acquired can be shown. The trigger is placed at the end of memory.

TS, FS, TD, TL, PP - *halt*

halt_repetitive_execution *halt*

Command used to increase resolution of duration triggers, increase sequencing capability, and to provide event-count triggering. Execution finishes when your specified condition is found.

PP - *halt_repetitive_execution when_time_x_o greater_than 40 usec*

hertz *Hz*

Used with the *sample* command to set the frequency multiplier of the analyzer sample clock.

TS, TD, TL - *sample rate_is 2 hertz*

indicate *indicate*

Command used to call for display of the time interval measured between the mark_x and mark_o samples, or to indicate the number of mark_<abcd> samples between the mark_x and mark_o samples. Additionally, in the Timing Diagram, the command can be used to call for display of the state of each trace at the cursor position. The arrowheads at the end of each trace label are replaced with h, l, m, or g to show high, low, middle, or glitch.

TD - *indicate levels_at_cursor*

TD, TL - *indicate time_interval_x_o*

indicator *indicator*

Used with the *magnify* command to turn on or off the horizontal bar, above each trace, which shows the region that will be displayed at the next higher level of magnification.

TD - *magnify indicator on*

in_bin *in_bin*

Used with the *display* command to define that the preceding label will be displayed in binary format.

TL - *display POD1 in_bin*

in_dec *in_dec*

Used with the *display* command to define that the preceding label will be displayed in decimal format.

TL - *display POD1 in_bin*

in_hex *in_hex*

Used with the *display* command to define that the preceding label will be displayed in hexadecimal format.

TL - *display POD1 in_hex*

in_oct *in_oct*

Used with the *display* command to define that the preceding label will be displayed in octal format.

TL - *display POD1 in_oct*

kilohertz *KHz*

Used with the *sample* command to set the frequency multiplier of the analyzer sample clock.

TS, TD, TL - *sample rate_is 20 kilohertz*

leaving *leaving*

Used with the *trigger* command to define the trigger as the point where a particular pattern changes from true to not-true. The true condition must precede the not-true; if the condition is always not-true, triggering will not occur. If the trigger drives the rear panel BNC connector, the signal there will be a narrow pulse. Note that an ANDed condition will trigger when the entire condition goes false, e.g.- all elements are true and then at least one element goes false to cause the trigger. Used with the *find* and *mark* commands to define the point to be found (and/or marked) as the point where a particular pattern changes from true to not-true. The true condition must precede the not-true; if the condition is always not-true, it will not be found (or marked).

TS - *trigger on leaving POD1 .0 = 0 and POD1 .1 = 0*
(note that *trigger on leaving POD1 = 0XXH* will never trigger)

TD, TL - *find leaving POD1 .6 = 1 and POD1 .5 = 0 thru mark_x*

TD, TL, PP - *mark a on_all_occurrences_of POD1 .0 = 1*

less_than *less*

Used with the *trigger* command to time qualify the trigger pattern, allowing as triggers only those patterns which exist for a time shorter than the specification. The trigger position will indicate where the time specification was satisfied. If the trigger is output to the rear panel BNC connector, the signal there will be a narrow pulse at the time that the pattern goes false. The available time range is from 1 msec to 20 nsec for asynchronous triggers, and from 1 msec max to two sample periods min for synchronous triggers. Used with the *find* command to define the point to be found as the point where a particular pattern remains true for less than the specified time period. The analyzer can only find the specified condition within the range of memory you designate.
Used with the *halt_repetitive_execution* command to designate a number of marks or a period of time which, when not met, will cause generation of the halt command for the repetitive execution in progress.

TS - *trigger on less_than 50 nsec_of POD1 = 0*

TD, TL - *find less_than 30 nsec_of POD1 .6 = 1 and POD1 .5 = 0 thru mark_x*

PP - *halt_repetitive_execution when_marks_x_o less_than 40* *levels__at__cursor* *levels*

Used with the *indicate* command to display the state of each trace at the cursor position. The arrowheads beside each trace label are replaced with h, l, m or g to show high, low, middle or glitch.

TD - *indicate levels__at__cursor*

load__from *load__from*

Used with the *configure* command to restore the Trace, Format, and Post_process Specifications and Timing Diagram and Trace List formats to those contained in a trace file saved previously. The data obtained during a previous measurement can also be restored to the trace memory if it was stored along with the configuration.

TS, FS, TD, TL, PP - *configuration load__from FILENAME:USERID*

logic__polarity *polarity*

Used with the *threshold* command to set the sense of a "1" as more positive or more negative than the probe threshold voltage for positive-true or negative-true, respectively. This allows trigger definitions to be made in terms of "1's" and "0's" independant of the volt-sense of the lines being measured. Trace List values will reflect these definitions but the Timing Diagram will show only positive-true voltage comparisons in oscilloscope fashion to prevent ambiguity.

FS - *threshold pod__1 all__bits ttl logic__polarity negative__true*

magnify *magnify*

Command used to change the time per division of the horizonal axis, or to turn on or off the bar which indicates the region that can be viewed at the next higher level of magnification.

TD - *magnify x10*

mark *mark*

Command used to assign identifiers to samples in the trace memory for use in making measurements between samples, for providing quick returns to marked samples, or to turn this capability on or off.

TD, TL, PP - *mark__x*

marked *marked*

Used with the *process* command to bring only the marked samples to the screen.

TL - *process__for__data marked*

mark__a *mark__a*

Used with the *find* command to bring any location marked with *a* to the display, centered if possible, and place the cursor on it (*mark__b*, *mark__c*, and *mark__d* work the same as *mark__a*). Used with the *halt__repetitive__execution* command to designate a member of a sequence of occurrences which, when found in order, will halt the repetitive executions in progress.

TD, TL - *find mark__a*

PP - *halt__repetitive__executions when__sequence__x__o mark__a then mark__a then__not mark__c*

mark__b *mark__b*

Refer to *mark__a* .

mark__c *mark__c*

Refer to *mark__a* .

mark__d *mark__d*

Refer to *mark__a* .

mark__names *mark__name*

Used with the *display* command to get a column on screen that shows the mark names you assigned in the Post_process Specification.

TL - *display POD1 in__hex then mark__names*

mark__o *mark__o*

Used with the *find* command to bring the location marked with *o* to the display, centered if possible, and put the cursor on that sample.

TD, TL - *find mark__o*

mark__x *mark__x*

Used with the *find* command to bring the location marked with *x* to the display, centered if possible, and put the cursor on that sample.

TD, TL - *find mark__x*

maximum__and__minimum *max__min*

Used with the *indicate* command to obtain two additional counts on the display, one indicating the largest number and the other indicating the smallest number ever measured during the series of repetitive executions of the selected measurement.

TD, TL - *indicate number__of__marks__x__o maximum__and__minimum* *mean_and_standard_deviation* *mean_stdv*

Used with the *indicate* command to obtain two additional counts on the display relating to repetitive executions of a selected measurement: the *mean* number indicates the average count obtained during the measurement, and the *stdv* number indicates the standard deviation from the mean value.

TD, TL - *indicate number_of_marks_x_o
mean_and_standard_deviation*

megahertz *MHz*

Used with the *sample* command to set the frequency multiplier of the analyzer sample clock.

TS - *sample rate_is 200 megahertz*

mode_is *mode*

Command used to change the operating modes of the Timing Analyzer to Dual Threshold, Fast Sample, Glitch Capture, or Wide Sample Mode. In Dual Threshold Mode, the lower numbered four inputs on each probe pod are compared to two voltages and displayed as three-level traces. The Fast Sample mode samples the lower four inputs at a 400-MHz rate. In Glitch Capture Mode, the occurrence of multiple data transitions on the lower inputs is recorded to provide an indication of unsampled activity on inputs at low sample rates when an overview is desired. Wide Sample Mode uses all eight probe inputs for maximim input width.

TS, FS - *mode_is dual_threshold*

modify *modify*

Used with the *trigger*, *trigger enable*, *display*, and *process* commands to recall the current specification to the command line for editing, to avoid extensive re-entry typing.

TS - *trigger modify*
TS - *trigger enable modify*
TD, TL - *display modify*
TL, PP - *process_for_data modify*
FS - *modify LABEL*

msec *msec*

Used with the *sample* command to set the period multiplier of the analyzer sample clock. Used with the *halt_repetitive_execution* command to identify the time period to be referenced when the analyzer determines whether or not to stop repetitive executions in progress.

TS, TD, TL - *sample period_is 25 msec*
PP - *halt_repetitive_executions when_time_x_o
greater_than 35 msec*

msec_after *ms_after*

Used with the *trigger* command to enter a time multiplier specifying the delay from when the trigger pattern definition is satisfied to when the trigger occurs. The delay range is from 2 to 32,000,000 sample periods, or from 1 to 32767 msec, whichever is less.

TS - *trigger 123 msec_after POD1 = 0*

msec_of *msec_of*

Used with the *trigger* command to enter the multiplier for time qualification on trigger patterns.

TS - *trigger on greater_than 1 msec_of POD1 = 0*

negative_true *negative*

Used with the *define* command to specify that for this label, a voltage more negative than the threshold voltage will be considered a "1" in the Trace Specification *trigger* definition and in the Trace List.

FS - *threshold pod_1 all_bits ttl logic_polarity negative_true*

neg_transition_on *negative*

Used with the *process* command to specify that a transition on the selected probe line will serve as the reference event.

TL, PP - *process_for_data sampled 3 samples_after neg_transition_on POD1 .6*

none *none*

Used with the *compare_file_is* command to cancel an existing compare file specification.

PP - *compare_file_is none*

nsec *nsec*

Used with the *sample* command to set the period multiplier of the analyzer sample clock. Used with the *halt_repetitive_execution* command to identify the time period to be referenced when the analyzer determines whether or not to stop repetitive executions in progress.

TS, TD, TL - *sample period_is 25 nsec*

PP - *halt_repetitive_executions when_time_x_o greater_than 200 nsec*

nsec_after *ns_after*

Used with the *trigger* command to enter a time multiplier specifying the delay from when the trigger pattern definition is satisfied to when the trigger occurs. The delay range is from 2 to 32,000,000 sample periods, or from 1 to 32767 nsec, whichever is less.

TS - *trigger 100 nsec_after POD1 = 0* *nsec_of* *nsec_of*

Used with the *trigger* command to enter the multiplier for time qualification on trigger patterns.

TS - *trigger on greater_than 100 nsec_of POD1 = 0*

number_of_marks_x_o *marks*

Used with the *indicate* command to obtain a count on the display of the number of mark_<abcd> samples between the mark_x sample and the mark_o sample in memory.

TD, TL - *indicate number_of_marks_x_o*

o *o*

Used with the *mark* command to identify a particular sample in trace memory for measurements between samples or for ease of finding that sample.

TD, TL - *mark o*

off *off*

Used with the *magnify*, *mark*, and *indicate* commands to turn off the magnification indicator bar or level indicator characters, and to turn marks *on* and *off*. Used with the *process* command to turn off display qualification, allowing display of all samples in memory. Used with the *halt_repetitive_execution* command to specify that repetitive executions of the measurement will not be halted.

TD - *magnify indicator off*
TD - *indicate levels_at_cursor*
TD, TL - *mark x off*
TL, PP - *process_for_data off*
PP - *halt_repetitive_executions off*

on *on*

Used with the *trigger* command to define a pattern to be used for triggering, with *magnify* and *indicate* commands to enable the magnification indicator bar and level indicator characters, and with the *mark* command to enable the marks.

TS - *trigger on POD1 = 4AH*
TD - *magnify indicator on*
TD - *indicate levels_at_cursor*
TD, TL, PP - *mark x on*

on_all_occurrences_of *on_all*

Used with *mark_<abcd>* to specify events to be identified with the corresponding marks.

TD, TL, PP - *mark a on_all_occurrences_of entering POD1 .6 = 1 after trigger* *on_cursor* *on_cursor*

Used with the *mark_x* and *mark_o* commands to assign a mark to follow the cursor position wherever it moves.

TD, TL, PP - *mark o on_cursor*

on_first_occurrence_of *on_first*

Used with the *mark_x* and *mark_o* commands to define the event to be identified with the corresponding mark.

TD, TL, PP - *mark_x on_first_occurrence_of entering POD1 .6 = 1 after trigger*

on_internal_nondelayed_trigger *on_triggr*

Used with the *assert* command to define that the BNC port will be driven when the trigger condition occurs. Internal time or Inter Module Bus clock delays are not applied.

TS - *assert bnc_port_4 on_internal_nondelayed_trigger*

on_sample *on_sample*

Used with the *mark_x* and *mark_o* commands to place an identifying mark on a numbered sample. The trigger position is defined as sample 0 and all samples are numbered + or - with respect to the trigger location.

TD, TL, PP - *mark o on_sample 256*

on_trigger *on_triggr*

Used with the *trigger* command to define when the Low Trigger Enable line in the Inter Module Bus will be driven. Used with the *mark_x* and *mark_o* commands to place a mark on the trigger position.

TS - *trigger enable driven on_trigger*

TD, TL, PP - *mark x on_trigger*

or *or*

Used with the *trigger* command in a 16-channel timing analyzer to specify alternate conditions for the trigger condition.

TS - *trigger on POD1 = 0 or POD2 = 0*

or_on *or_on*

Used with the *trigger* command in Glitch Mode to define additional inputs on which glitches can be recognized as triggers.

TS - *trigger on any_glitch on POD1. 0 or_on POD1. 1*

percent_after_start_of_trace *after*

Used with the *trigger* command to locate the relative position of the trigger within the acquisition memory in 5% increments after the beginning of memory.

TS - *trigger position_is 20 percent_after_start_of_trace* *percent__before__end__of__trace* *before*

Used with the *trigger* command to locate the relative position of the trigger within the acquisition memory in 5% increments before the end of memory.

TS - *trigger position__is 20 percent__before__end__of__trace*

period__is *period*

Used with the *sample* command to define the time interval between samples. The allowed range is 5 nsec (10 nsec in Glitch Mode) to 500 msec.

TS, TD, TL - *sample period__is 50 nsec*

pod__1 *pod__1*

Used with the *threshold* command to define the pod to which the threshold voltage definition applies. Used with the *display* command to format the sequence of traces on screen with labels showing probe inputs. The probe connected to the Acquisition board in the lower numbered mainframe slot in a 16-channel analyzer is defined as probe pod #1. In an 8-channel analyzer, it is the only probe pod. While the POD1 label is redefinable, *pod__1* is not.

FS - *threshold pod__1 all__bits ecl*
TD, TL - *display default__bits__in pod__1*

pod__1__bit *pod__1__bit*

Used with the *define* command to establish a mapping between a label and probe pod #1 inputs.

FS - *define NEWLABEL pod__1__bit 0 width 3*

pod__2 *pod__2*

Used with the *threshold* command to define the pod to which a threshold voltage applies. Used with the *display* command to format the sequence of traces on screen with labels showing probe inputs. The probe that connects to the Acquisition Board in the higher numbered mainframe slot in a 16-channel analyzer is defined as probe pod #2. In an 8-channel unit, *pod__2* is undefined. While the label POD2 is redefinable, *pod__2* is not.

FS - *threshold pod__2 all__bits ecl*
TD, TL - *display default__bits__in pod__2*

pod__2__bit *pod__2__bit*

Used with the *define* command to establish a mapping between a label and the inputs of probe pod #2 in a 16-channel analyzer.

FS - *define NEWLABEL pod__2__bit 0 thru pod__1__bit 6*

position__is *position*

Used with the *trigger* command to locate the relative position of the trigger within the trace memory.

TS - *trigger position__is start__of__trace* *positive_true* *positive*

Used with the *define* command to specify that for this label, a voltage more positive than the threshold voltage will be considered a "1" for trigger definition in the Trace Specification, and in the Trace List.

FS - *threshold pod_1 all_bits ttl logic_polarity positive_true*

post_process_specification *postspec*

Used with the *show* command to select display of the Post_process Specification.

TS, FS, TL, TD - *show post_process_specification*

pos_or_neg_transition_on *pos_neg*

Used with the *process* command to specify that either a positive-going or negative-going transition on the selected probe line will serve as the reference event.

TL, PP - *process_for_data sampled 3 samples_after pos_or_neg_transition_on POD1 .6*

pos_transition_on *positive*

Used with the *process* command to specify that a positive-going transition on the selected probe line will serve as the reference event.

TL, PP - *process_for_data sampled 2 samples_before pos_transition_on POD1 .7*

printer *printer*

Used with the *copy* command to define the printer as the destination device to which copies of Specifications will be sent.

TS, FS, PP, TD, TL - *copy all_specifications to printer*

process_for_data *process*

Command used for display qualification. It allows only samples that meet your specification to be displayed on screen.

PP, TL - *process_for_data sampled 2 samples_before pos_transition_on POD1 .6*

rate_is *rate*

Used with the *sample* command to define the frequency at which samples are taken. The allowed range is from 2 Hz to 200 MHz (100 MHz in Glitch Mode).

TS, TD, TL - *sample rate_is 200 megahertz*

received *received*

Used with the *trigger* command to specify the High TRigger or Low Trigger Enable line in the Intermodule Bus as the trigger source, or for enabling the internal trigger, respectively. The HTR and LTE cannot be received simultaneously.

TS - *trigger received*

TS - *trigger enable received*

received_or_driven *rec/drive*

Used with the *trigger* command to both receive the High TRigger line in the Intermodule bus as a trigger source or if the internal trigger condition is satisfied first, to trigger and also drive this line, i.e. this is the logical OR of an internal and an external trigger.

TS - *trigger received_or_driven*

relative *relative*

Used with the *display* command to define that the time displayed will be differential time between the samples on each line.

TL - *display POD1 then time_count relative*

rename *rename*

Command used to alter an existing label by changing its name. The change will be reflected in all other specifications and displays.

FS - *rename OLDLABEL to NEWLABEL*

repetitively *repeat*

Used with the *execute* command to require that another measurement begin as soon as the current measurement is complete. Measurements will continue until the *halt* key is pressed.

TS, FS, PP, TD, TL - *execute repetitively*

restart *restart*

Used with the *trigger* command to enable control of Timing Analyzer measurements by another module on the Intermodule Bus. This option is only available after *trigger enable received* has been specified. When Trigger Enable goes false, the Timing Analyzer resets and begins a new measurement; when Trigger is true, the measurement in progress is allowed to complete, or the data in memory is retained.

TS - *trigger enable restart*

sample *sample*

Command used to set the frequency or period at which data will be acquired and stored by the analyzer. The allowed range is from 2 Hz/500 msec to 200 MHz/5 nsec (2 Hz/500 msec to 100 MHz/10 nsec in Glitch Mode). The default is the fastest available sample rate in each mode.

TS, TD, TL - *sample rate_is 200 megahertz*

sampled *sampled*

Used with the *process* command for display qualification. This qualification allows display of only those samples that occurred at the specified locations.

TL, PP - *process_for_data sampled 2 samples_after neg_transition_on POD1 .6* *samples_after* *after*

Used with the *process* command to qualify display of the selected sample after the designated occurrence.

TL, PP - *process_for_data sampled 2 samples_after neg_transition_on POD1 .6*

samples_before *before*

Used with the *process* command to qualify display of the selected sample before the designated occurrence.

TL, PP - *process_for_data sampled 2 samples_before neg_transition_on POD1 .2*

save_in *save_in*

Used with the *configure* command to store the current Trace, Format, and Post_process Specifications and the Timing Diagram and Trace List formats (and trace memory content, if desired), in a Trace file.

TS, FS, PP, TD, TL - *configuration save_in FILENAME:USERID with_data*

sec *sec*

Used with the *halt_repetitive_execution* command to identify the time period to be referenced when the analyzer determines whether or not to stop the repetitive executions in progress.

PP - *halt_repetitive_executions when_time_x_o greater_than 10 sec*

sec_after *sec_after*

Used with the *trigger* command to enter a time multiplier specifying the delay from when the trigger pattern definition is satisfied to when the trigger occurs. The delay range is from 2 to 32,000,000 sample periods, or 1 to 32767 sec, whichever is less.

TS - *trigger 1 sec_after POD1 = 0*

show *show*

Command used to change the display to another specification or to the Timing Diagram or Trace List.

TS, FS, TD, PP - *show trace_list*
TS, FS, PP, TL - *show timing_diagram*
TS, PP, TD, TL - *show format_specification*
PP, FS, TD, TL - *show trace_specification*
TS, FS, TD, TL - *show post_process_specification*

sixteen_channels *16_chanls*

Used with the *diagram* command to set the vertical display format to sixteen traces on screen.

TD - *diagram eight_channels*

start *start*

Used with the *copy* command to indicate that the range of information to be transfered begins at the start of the trace and ends at the present cursor position.

TL - *copy trace_list thru start*

start_of_trace *start*

Used with the *trigger position_is* command to specify the trigger position is 30 samples after the start of the trace (70 in Fast Sample Mode).

TS - *trigger position_is start_of_trace*

then *then*

Used with the *display* command to add another trace when formatting the display sequence. Used with the *halt_repetitive_execution* command to add another event to a sequence of events which must be found to generate the halt command for a series of repetitive executions.

TD, TL - *display POD1. 0 then POD1. 2*

PP - *halt_repetitive_execution when_sequence_x_o mark_a then mark_b then mark_c*

then_not *then_not*

Used with the *halt_repetitive_execution* command to include a not-occurrence in a sequence of occurrences which must be found to generate the halt command for a series of repetitive executions.

PP - *halt_repetitive_execution when_sequence_x_o mark_a then_not mark_d then mark_a*

threshold *threshold*

Command to set the comparison voltage(s) that the analyzer uses to determine logic levels of the probe input signals. The allowed range is from -10.0V to +10.0V in 0.1V steps. Default is ttl (+1.4V).

FS - *threshold all_pods all_bits ecl*

thru *thru*

Used with the *define* command to specify the highest numbered bit in a range of bits to be included in the definition of a label. Used with the *copy* command to indicate the end of a range of lines to be transferred to a file or device.

FS - *define LABEL pod_1_bit 0 thru pod_1_bit 4*

TD, TL - *copy trace_list thru end to printer*

time_count *time_cnt*

Used with the *display* command to include a column showing either relative or absolute times.

TL - *display POD1 then time_count relative*

time_interval_x_o *time_x_o*

Used with the *indicate* command to obtain a count on the display of the time interval between the mark_x sample and the mark_o sample in memory.

TD, TL - *indicate time_interval_x_o* *timing_diagram* *diagram*

Used with the *show* command to bring the Timing Diagram on screen for viewing the results of a trace or formatting the display sequence.

TS, FS, PP, TL - *show timing_diagram*

to *to*

Used with the *rename* command to specify a new name to be used in place of an existing label name. Used with the *copy* command to identify the destination file or device to which a specification is to be copied.

FS - *rename OLDLABEL to NEWLABEL*
TS, FS, PP, TD, TL - *copy all_specifications to printer*

trace_list *tracelist*

Used with the *show* command to indicate that the trace list will be brought to the display. Used with the *copy* command to indicate that the Trace List Display is to be transferred to a destination file or device.

TS, FS, PP, TD - *copy trace_list*
TL - *copy trace_list to FILENAME:USERID*

trace_specification *tracespec*

Used with the *show* command to bring the Trace Specification onto the screen for setting triggers, sample rate, and interactions with other modules.

TS, PP, TD, TL - *show trace_specification*

trigger *trigger*

Command to define the pattern(s) of logic levels to be found on the probe inputs, and to include time or event delays, Intermodule Bus signal interactions, and relative memory positioning (which can be used for ending an execution). The default is a "don't-care" trigger. Used with the *find* command to specify that the trigger sample is to be placed at the center of the screen, if possible, and to locate the cursor at that point.

TS - *trigger on any_thing*
TD, TL - *find trigger*

ttl *ttl*

Used with the *threshold* command to specify the comparison voltage for the probe inputs is the middle of the LS TTL voltage range, +1.4V. In Dual Threshold Mode, the upper and lower threshholds are set to +2.0V and +0.8V, respectively.

FS - *threshold pod_1 all_bits ttl*

usec *usec*

Used with the *sample* command to set the period multiplier of the analyzer sample clock. Used with the *halt__repetitive__execution* command to identify the time period to be referenced when the analyzer determines whether or not to stop a series of repetitive executions.

TS, TD, TL - *sample period__is 25 usec*
PP - *halt__repetitive__executions when__time__x__o*
greater__than 200 usec

usec__after *us__after*

Used with the *trigger* command to enter a time multiplier for specifying delay between recognition of the trigger pattern and identification of the trigger sample. The delay range is from 2 to 32,000,000 sample periods, or from 1 to 32767 msec, whichever is less.

TS - *trigger 123 msec__after POD1 = 0*

usec__of *usec__of*

Used with the *trigger* command to enter a multiplier for time qualification of trigger patterns.

TS - *trigger on greater__than 1 usec__of POD1 = 0*

volts *volts*

Used with the *threshold* command to enter a voltage value for comparison with the voltage of each probe input when determining signal logic levels.

FS - *threshold pod__1 all__bits 5.0 volts*

when__marks__x__o *marks*

Used with the *halt__repetitive__execution* command to specify a number of *mark__<abcd>* samples to be counted between the mark_x and mark_o samples, and when the count exceeds the specified number of samples, the timing analyzer will generate the *halt* command for repetitive executions.

PP - *halt__repetitive__execution when__marks__x__o*
greater__than 26

when__runs__equals *runs*

Used with the *halt__repetitive__execution* command to specify a number of runs to be included in a measurement made up of a series of repetitive executions.

PP - *halt__repetitive__execution when__runs__equals 99*

when__sequence__x__o *sequence*

Used with the *halt__repetitive__executions* command to specify a sequence of occurrences (identified by mark_<abcd> designators) which, when found, will cause the timing analyzer to generate the *halt* command for repetitive executions.

PP - *halt__repetitive__execution when__sequence__x__o*
mark__a then mark__b then__not mark__c

when__time__x__o *time*

Used with the *halt__repetitive__executions* command to specify a period of time to be measured between the mark_x and mark_o samples which, when found, will cause the timing analyzer to generate the *halt* command for repetitive executions.

PP - *halt__repetitive__execution when__time__x__o greater__than 40 usec*

x *x*

Used with the *mark* command to identify a particular sample within the trace memory for making measurements between samples or for returning to that location easily.

TD, TL - *mark x*

x1 *x1*

Used with the *magnify* command to select the horizontal magnification that shows the entire memory contents on screen (half contents in Fast Sample Mode). In effect there are four-hundred samples per division. If there are transitions which are too close together to be be displayed at this horizontal resolution, they are shown with a Glitch symbol (broken vertical bar).

TD - *magnify x1*

x10 *x10*

Used with the *magnify* command to select the horizontal magnification that shows about 10% of the memory contents on screen (5% in Fast Sample Mode). In effect, forty samples are shown per division. If there are transitions too close together to be displayed at this horizontal resolution, they will be shown with a Glitch symbol (broken vertical bar).

TD - *magnify x10*

x100 *x100*

Used with the *magnify* command to select the horizontal magnification that shows about 1% of the memory contents on screen (0.5% in Fast Sample Mode), at four samples/division.

TD - *magnify x100*

when__greater__than *when*

Used with the *trigger* command in 16-channel analyzers to define a second condition which must be simultaneously true when triggering on a first condition. The second condition requires a pattern that must be true for more than a specified period of time. When the first condition in the trigger specification includes a glitch, the entire specfication will be synchronous and the time will be constrained to be two or more sample periods.

TS - *trigger on entering POD1 = 0 when__greater__than 1 usec__of POD2 = 0* *wide_sample* *wide*

Used with the *mode_is* command to set the operating mode of the analyzer to sample data on all eight inputs on each probe pod.

TS, FS - *mode_is wide_sample*

width *width*

Used with the *define* command to specify the number of bits to be included in a label. The bits start with the one specified and include as many higher numbered bits as specified, limited to the bits available in the hardware.

FS - *define NEWLABEL pod_1_bit 3 width 5*

with *with*

Used with the *trigger* command in the Glitch Mode to require simultaneous occurrence of a pattern and one glitch to satisfy the trigger condition.

TS - *trigger on any_glitch on POD1. 0 with POD1 = 0*

with_data *with_data*

Used with the *configure* command to store the content of the trace memory along with the trace file containing the Trace, Format, and Post_process Specifications, and the Timing Diagram and Trace List formats.

TS, FS, TD, TL, PP - *configuration save_in FILENAME:USERID with_data*

write_protect *protect*

Used with the *configure* command to prevent overwriting a measurement configuration file containing Trace, Format, and Post_process Specifications, and Trace List and Timing Diagram formats (with or without trace memory data).

TS, FS, TD, TL, PP - *configuration save_in FILENAME:USERID write_protect*

Appendix C

ERROR MESSAGES

INTRODUCTION

The following messages are displayed on screen to provide an indication of operating status. Some of these also advise of improper operating conditions or invalid entry on the command line.

"and" is not possible, all bits are already specified - Displayed when you try to enter a trigger specification which uses the same bit or bits in two or more labels. Your specification must specify only one entry for each bit.

"and" is not possible, both pods are already used - Displayed when your trigger specification is an ANDing of values in which at least one of the pods is mentioned more than one time. A "<>" condition on a given pod does not allow any additional definitions on that pod.

Attempt to define too many labels - Displayed when you try to define more labels than the Timing Analyzer can use. The Timing Analyzer can operate with up to 31 labels defined.

BNC port 4 already driven by another module - Displayed when you try to drive BNC port 4 from the timing analyzer and it is already being driven by another module.

Cannot save configuration into compare file - Displayed when you try to save the instrument configuration into the file that is presently designated as the compare_file.

Checksum error, file is corrupt - Displayed when a file is loaded from memory and the checksum on the file does not agree with the checksum calculated by the Timing Analyzer during the load operation.

Compare file fills the symbol table - Displayed when the configuration named as the compare file has enough labels assigned in it to fill the capacity of the Timing Analyzer (31 labels, maximum) when combined with the labels in the current configuration.

Compare file spec does not agree with hardware - Displayed when the data in the trace memory was obtained by using a different sample rate, trigger position, or mode than was used to obtain the data in the compare file.

Command is not valid during an execution - Displayed when you try to enter a specification command which is invalid during an execution.

Data is not present in file - Displayed when you try to specify a file that was not stored with data to be used as a compare_file.

Data is not present in hardware - Displayed when you command the Timing Analyzer to save its configuration with_data, and there is no data in the trace memory.

Data label width exceeds instrument's capabilities - Displayed when you try to create a label having more probe bits than are available in the configuration of your Timing Analyzer.

Data label <YOUR LABEL> bits are already specified - Displayed when your trigger specification includes the same bit or bits in two or more labels in your trigger specification.

Data label <YOUR LABEL> is not a valid entry - Displayed when your trigger specification includes a label which is not valid for the current mode which is set up.

Data label <YOUR LABEL> specified bit does not exist - Displayed when your trigger specification includes a label-bit that is not defined in the Format Specification.

Disc 0 is down - Displayed when you try to write a file to or load a file from disc drive 0, and the disc drive is not operating.

Disc 0 is full - Displayed when you try to save a display or configuration on disc 0 when it has no storage space left.

Disc 0 is full, writing to disc 1 - Displayed when you try to save a display or configuration and the Timing Analyzer finds disc 0 full but space available on disc 1. It automatically switches the save function to disc 1 and processes your save command.

Disc 1 is down - Displayed when you try to write a file to or load a file from disc drive 1, and the disc drive is not operating.

Disc 1 is full - Displayed when you try to save a display or configuration on disc 1 when it has no storage space left.

Duration is greater than trace memory - Displayed when the entered duration is greater than the duration of the entire trace memory.

Duration is less than sample period - Displayed when entered duration is less than the sample period. The timing analyzer cannot find durations which are less than the sample period.

Execution error - Displayed when measurement_system module cannot execute all previously configured modules.

Expression too long, shorten mark and trigger expressions - Displayed when the combination of all the trigger definitions and mark definitions are too long. Shorten or remove unused definitions.

File exists, but is not a timing analysis file - Displayed when you try to save a configuration or load a configuration from a filename that already is used for storage of a configuration of some other instrument, such as a state/software analyzer.

File is write protected - Displayed when you try to save a configuration under a file name that already contains a write-protected configuration.

File version <FILE NO.> is incompatible with software version 2310 - Displayed when you attempt to use a timing analysis configuration file with a later date (future version) for comparison with the current version of software.

Filling trace memory - Displayed when the timing analyzer is filling its trace memory in order to satisfy the prestore requirements. Triggers are not acknowledged during this time.

"followed_by" is not possible, both pods are used - Displayed when your trigger specification requires a condition to be found followed by another condition, and both conditions include the same probe bit or bits. The timing analyzer can trigger on a condition found on one probe pod bit or bits followed_by a condition found on a different probe pod bit or bits.

Halt_rept_exec duration is greater_than trace memory - Displayed when the entered duration is greater than the duration of the trace memory.

Halt_rept_exec duration is less_than sample period - Displayed when the entered duration is less than the sample period.

Hardware error detected - Displayed when the timing analyzer loader cannot load software into the timing analysis boards.

Hardware error in graphics printer board - Displayed when the graphics printer board has a hardware malfunction.

Hpib only possible in stand-alone mode - Displayed when you try to specify the input or output device as HP-IB port and you are operating in a cluster environment.

Invalid compare file, measurement was halted - Displayed when you try to define a halted measurement as a compare file.

Labels used in any other specification cannot be deleted - Displayed when you try to delete a label from this specification that is used as a parameter in some other specification in the Timing Analyzer.

Lower threshold must be less than upper threshold - Displayed during Dual_Threshold mode when you try to enter a specification which places the lower threshold at a voltage more positive than the upper threshold.

Mark "after condition" is invalid - Displayed when you try to mark a condition and you specify an "after" condition using the same mark ("mark x ... after mark_x" is invalid).

Marking complete, marking limit of 511 exceeded - Displayed when the timing analyzer finishes marking the data and has found more than 511 events that meet specifications of mark_<abcd>.

Measurement in process, cannot alter specification - Displayed when you try to modify any of the measurement parameters while the Timing Analyzer is involved in making a measurement.

Measurement__system error - Displayed when the measurement system software is not present.

Mode is not glitch capture - Displayed when you try to enter a trigger specification requiring glitch detection and the Timing Analyzer is not in the glitch mode of measurement.

Module not involved - Displayed when the timing analyzer module is not involved in the current measurement.

Multiple drivers - Displayed when multiple drives are set up to operate on a given intermodule bus line.

New label already exists - Displayed when you try to rename a label to a label which already exists.

"or" is not possible, both pods are already used - Displayed when your trigger specification is an ORing of values in which at least one of the pods is mentioned more than one time. An "OR" condition can only occur between pods.

"or__on" is not possible, all bits are already specified - Displayed when you try to enter a trigger specification which calls for finding a value on a bit or set of bits when your specification already includes values for all the bits under other labels.

"or__on" is not possible, both pods are already used - Displayed when you try to enter your trigger specification which requires a value to be found on a probe input and glitch detection on two or more probe pod inputs. Triggering can be specified to occur on a value ANDed with glitch detection on one probe input channel.

Pod__2 hardware is not present - Displayed when you try to enter a specification for a condition to be found on pod 2 in an eight-channel analyzer.

Processing complete, marking limit of 511 exceeded - Displayed when the timing analyzer completes processing the trace list and more than 511 mark_<abcd> events were found.

Process__for__data definition is invalid - Displayed when you change the mode or redefine the labels in such a way that the process_for_data definition is no longer valid.

Range must be greater than one sample - Displayed when you try to find an event and the range you specified for the timing analyzer to look in is only one sample wide.

Sample exceeds memory depth - Displayed when you enter a specification to process_for_data a number of samples "before" or "after" a transition and the number of samples is greater than the memory depth of the timing analyzer (4060 or 8140 samples).

Sample period is fixed at 2.5 nsec in fast sample mode - Displayed when you try to enter a different sample period specification while the Timing Analyzer is operating in the Fast Sample mode.

Sample rate is fixed at 400 MHz in fast sample mode - Displayed when you try to enter a different sample rate specification while the Timing Analyzer is operating in the Fast Sample mode.

Sample rate is too slow for synchronous trigger duration - Displayed when you try to enter a combinational glitch trigger definition where the sample point is longer than the trigger duration. Glitch triggering requires all data to be sampled before trigger duration detection circuitry.

Sequence limit is four marks - Displayed when you try to enter a halt_repetitive_execution sequence which is greater than four marks long.

Single bit data label must be entered - Displayed when you try to process_for_data relative to a transition on a label, and the label is defined mover more than one bit.

Single bit glitch label must be entered to use "with" - Displayed when you try to mark or find a value with a glitch and the glitch label is wider than one bit.

Specification does not agree with captured data - Displayed when you command the Timing Analyzer to save its configuration with_data, and you have changed the specification after to something different from what was used to capture the data.

Support for function is not present in hardware - Displayed when you try to drive or receive IMB signals which are not supported by another module.

Trace complete - Displayed when the timing analyzer has completed its trace in process.

Trace halted - Displayed when you halt the timing analyzer trace in process.

Trace in process - Displayed when the timing analyzer has found the condition that meets its trigger specification and is in process of completing its trace.

Trigger and trigger enable cannot both be received - Displayed when you try to enter a specification that requires trigger and trigger enable both to be received from an associated analysis module. Either one can be received, but the Timing Analyzer cannot receive both.

Trigger cannot be received or driven for restart function - Displayed when you try to enter the trigger enable restart command and a trigger specification of "trigger driven" or "trigger received" exists.

Trigger not found - Displayed when the "hold" signal is received (IMB trigger received) before the timing analyzer has found its timing analysis trigger.

Trigger pattern resource is used in trigger definition - Displayed when you try to enter a trigger enable definition using the pattern resource. Must have "trigger received" before a trigger enable definition using the pattern resource is valid.

Trigger pattern resource is used in trigger enable definition - Displayed when you try to enter a trigger definition using the pattern resource which is already used in the trigger enable definition. Change the trigger enable component to not include the pattern resource and then reenter your trigger definition.

Unknown file header error - Displayed when the software detects an unknown configuration file error.

Waiting for enable - Displayed when the timing analyzer is searching for the pattern to drive the trigger enable line.

Waiting for hold - Displayed when the timing analyzer is waiting to receive the "hold" signal (IMB trigger) from another analyzer to restart its measurement.

Waiting for trigger - Displayed when the timing analyzer has satisfied its prestore requirements but has not found the specified trigger condition. Note: the external enable must be true before trigger will be found.

"when_greater_than" is not possible, both pods are used - Displayed when your trigger specification requires a condition to be found ANDed with a duration of a value to be found, and both conditions include bits from the same probe pod.

"with" is not possible, need a single bit glitch label - Displayed when you try to enter a trigger specification which requires a value to be found on a probe input and glitch detection on two or more probe pod inputs. Triggering can be specified to occur on a value ANDed with glitch detection on one probe input channel.

-10.0 volts <= threshold <= 10.0 volts - Displayed when you try to specify a threshold voltage that is outside the range of threshold voltages that the Timing Analyzer can accept.

"<>" is not possible with "any_glitch on" - Displayed when your trigger specification ANDs glitch detection with a "not-equal-to" value to be found on a label. Glitch detection can only be ANDed with "equal-to" values on labels.

Appendix D

SYMBOLS USED ON TIMING ANALYZER DISPLAYS

The following symbols are used in displays presented by the Timing Analyzer. These provide information about the content of the displays.

t = trigger location

i = trigger location if supplied from an associated analyzer
module over the intermodule bus.

x = (1). location of sample marked 'mark_x'
(2). label and/or bit from a compare_file

o = location of sample marked 'mark_o'

a = location of sample marked 'mark_a'

b = location of sample marked 'mark_b'

c = location of sample marked 'mark_c'

d = location of sample marked 'mark_d'

m = multiple marks on a single sample

g = glitch found when taking the sample

'.' = the number that follows is a label channel number.

'--' = cursor position shown in trace list.

'^' = Portion of trace memory displayed on screen in
the timing diagram (indicates position of display
window in trace memory).

* = data from compare_file cannot be displayed because it
was obtained using a configuration whose attributes
are different from those used to obtain present data.

* = specified bit does not exist.

Index

FOLD HERE

HEWLETT
PACKARD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 1303 COLORADO SPRINGS, COLORADO

POSTAGE WILL BE PAID BY ADDRESSEE

HEWLETT-PACKARD
Attn: Logic Publications Dept.
Centennial Annex - D2
P.O. Box 617
Colorado Springs, Colorado 80901-0617

FOLD HERE

Your cooperation in completing and returning this form will be greatly appreciated. Thank you.

READER COMMENT SHEET

Part Number: 64601-90905

Your comments are important to us. Please answer this questionaire and return it to us. Circle the number that best describes your answer in questions 1 through 7. Thank you.

1. The information in this book is complete:

Doesn't cover enough (what more do you need?)	1	2	3	4	5	Covers everything

2. The information in this book is accurate:

Too many errors	1	2	3	4	5	Exactly right

3. The information in this book is easy to find:

I can't find things I need	1	2	3	4	5	I can find info quickly

4. The Index and Table of Contents are useful:

Helpful	1	2	3	4	5	Missing or inadequate

5. What about the "how-to" procedures and examples:

No help	1	2	3	4	5	Very helpful
Too many now	1	2	3	4	5	I'd like more

6. What about the writing style:

Confusing	1	2	3	4	5	Clear

7. What about organization of the book:

Poor order	1	2	3	4	5	Good order

8. What about the size of the book:

too big/small	1	2	3	4	5	Right size

Comments:

Particular pages with errors?

Name (optional):
Job title:
Company:
Address:

Note: If mailed outside U.S.A., place card in envelope. Use address shown on other side of this card.

FOLD HERE

HEWLETT
PACKARD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 1303 COLORADO SPRINGS, COLORADO

POSTAGE WILL BE PAID BY ADDRESSEE

HEWLETT-PACKARD
Attn: Logic Publications Dept.
Centennial Annex - D2
P.O. Box 617
Colorado Springs, Colorado 80901-0617

FOLD HERE

Your cooperation in completing and returning this form will be greatly appreciated. Thank you.

READER COMMENT SHEET

Part Number: 64601-90905

Your comments are important to us. Please answer this questionaire and return it to us. Circle the number that best describes your answer in questions 1 through 7. Thank you.

1. The information in this book is complete:

Doesn't cover enough (what more do you need?)	1 2 3 4 5	Covers everything

2. The information in this book is accurate:

Too many errors	1 2 3 4 5	Exactly right

3. The information in this book is easy to find:

I can't find things I need	1 2 3 4 5	I can find info quickly

4. The Index and Table of Contents are useful:

Helpful	1 2 3 4 5	Missing or inadequate

5. What about the "how-to" procedures and examples:

No help	1 2 3 4 5	Very helpful
Too many now	1 2 3 4 5	I'd like more

6. What about the writing style:

Confusing	1 2 3 4 5	Clear

7. What about organization of the book:

Poor order	1 2 3 4 5	Good order

8. What about the size of the book:

too big/small	1 2 3 4 5	Right size

Comments:

Particular pages with errors?

Name (optional):
Job title:
Company:
Address:

Note: If mailed outside U.S.A., place card in envelope. Use address shown on other side of this card.

64601-90905, JULY 1983
Replaces: 64601-90903, June 1982

HEWLETT
PACKARD

PRINTED IN U.S.A.